平成５年９月１日発行(毎月１回１日発行)　第30巻　第９号　通巻348号

DESIGN WAVE

1993
9

トランジスタ技術 別冊付録　デザインウェーブ

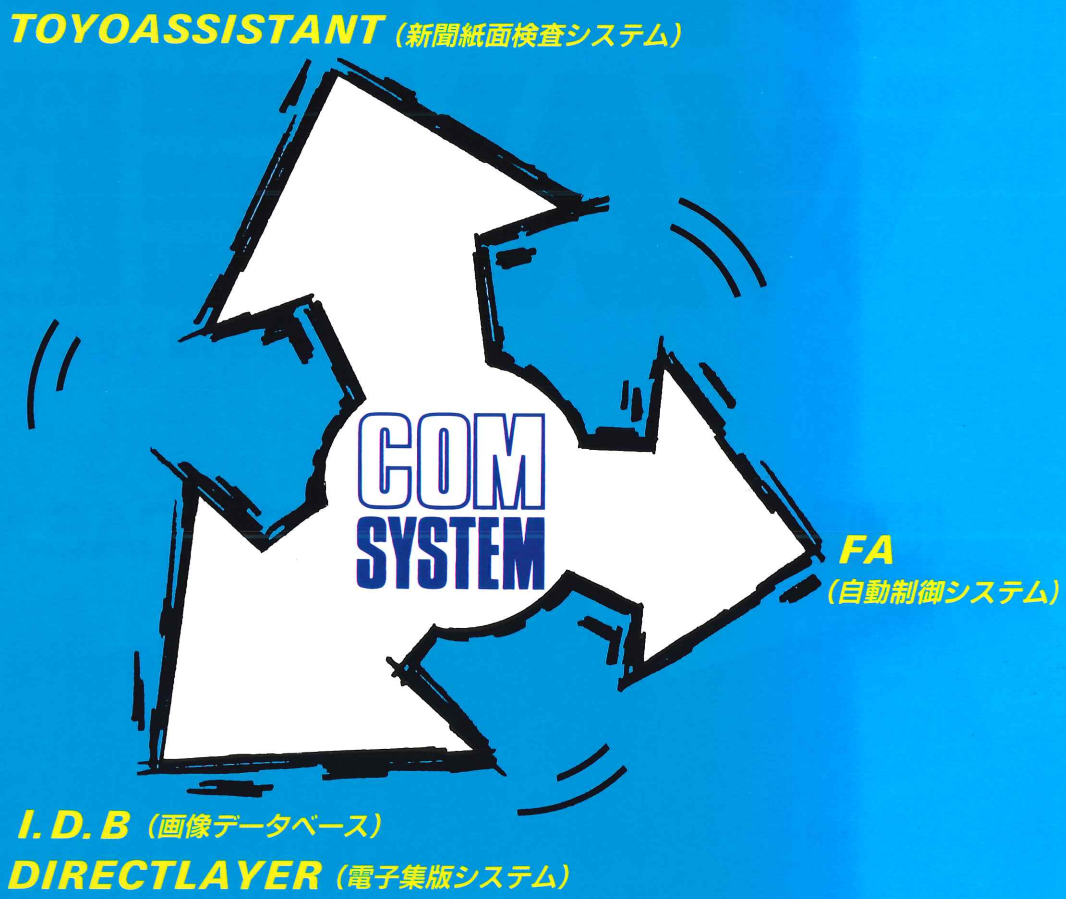
特集　ボード・レス設計を実現させるために　アナログ回路用CAE入門



- 連載　実用　ビヘービア・モデル作成術①
- 連載　例解　VHDLプログラミング①
- Why? What? コンパイラの仕様にない事項

トランジスタ技術
インターフェース 合同付録
Way

そのひらめき、隠さないでください。



せっかくのあなたのひらめき、アイディア。頭の中にしまったままではいけません。出して広げて、開発って仕事。本当はもっともっと楽しいはずですもんね。COM SYSTEMは画像処理・画像認識に特化した「研究開発先行」企業。画像処理・認識のハード＆ソフトを組み合わせてまったく未知のシステムを開発していきます。あなたのひらめきをどんどん取り入れて、世界を広げて行くCOM SYSTEM。隠さないで、しまわないで、あなたの「！」「？」カタチにしましょう。

職種	業務内容
❶ 画像処理システム開発	画像処理プロセッサのハードウェア開発設計。
❷ 画像認識システム開発	画像認識・解析システムのハードウェア開発設計。
❸ デジタル回路設計	画像認識・処理・解析プロセッサ及びカラーグラフィックシステム、FA等のハードウェア設計。

【会社概要】 ■**設立**／1979年2月1日 ■**資本金**／1億1750万円 ■**売上高**／10億5400万円 ■**従業員数**／65人 【募集要項】■**年齢**／大卒以上25歳〜33歳位迄 ■**給与**／経験・能力に応じ当社規定により優遇いたします：平成5年3月大卒初任給／205,000円（住宅手当1万円含む） ■**待遇**／昇給年1回、賞与年2回（夏期・年末）、家族手当、住宅手当、残業手当、役職手当、交通費全額支給 ■**福利厚生**／各種社会保険完備、財形貯蓄制度、団体生命保険加入 ■**休日休暇**／完全週休2日制、夏期休暇、年末年始休暇、有給休暇、特別休暇、（年間休日127日） ■**勤務時間**／9:00〜17:30（フレックスタイム制有）

【採用関係連絡先】TEL. **06・445・6057** 担当／森田

株式会社コムシステム

〒550 大阪市西区江戸堀1-25-22 新江戸堀ビル TEL 06-445-6057

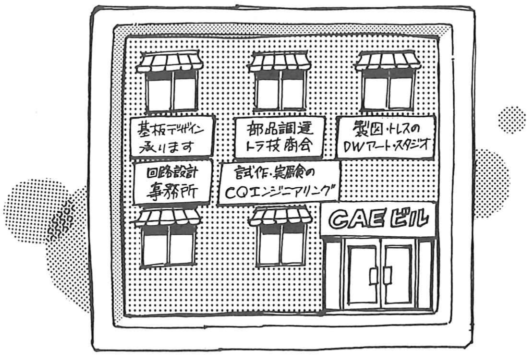
▶資料請求：TM01◀

DESIGN WAVE

1993
9

トランジスタ技術別冊付録 デザインウェーブ

特集 ボード・レス設計を実現させるための アナログ回路用CAE入門



CADやシミュレータは，すでに普及していると言ってしまってよいのでしょうか．言葉だけならもうすっかり知れ渡っているのでしょうが，実際に仕事のためのツールとして使っているエンジニアの比率はまだまだ低いのではないかと考えられます．

今回の特集では，エンジニアリング・ツールのこれからの普及にそなえ，CAD/シミュレータを包括し，それら以上に広い範囲をカバーするCAEという概念を紹介します．CAEは設計の高効率を実現するというだけでなく，アイデアの実現を助けるツールなのです．

CAEに限らず，CADやシミュレータをこれから導入しようと考えている読者にとってはツールを選択する際の参考になることでしょう．

CONTENTS

表紙デザイン/アイドマ・スタジオ　イラスト/神崎真理子



試作基板がなくなる時が来るのか？

アナログ回路用CAE入門

苗手 英彦

ワープロが普及してきたと同様に，回路図もパソコンなどのコンピュータを使用して作成することが今や当たり前のことになりました．

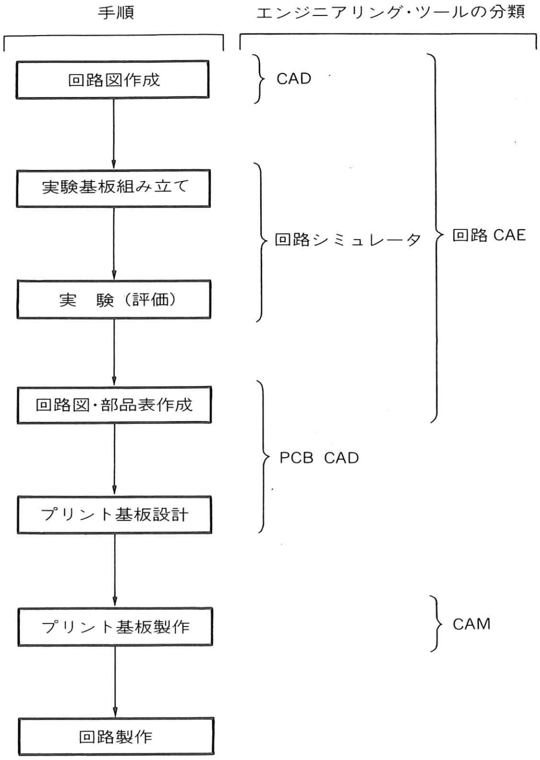
そして最近では，作成した回路図の検証(動作解析)もコンピュータで行われています．そこで本稿では，アナログ回路を中心とした回路の動作解析を行うことを目的とした，アナログ回路CAEについて解説します．

CAEとは？

従来，回路設計はおおよそ図1のような手順で，ほとんど手作業で行われてきました．

しかし，コンピュータの性能向上と普及にともない，その多くの部分でコンピュータの支援を受けるようになってきています．

〈図1〉回路設計の手順とエンジニアリング・ツールの範囲



それと言うのも，コンピュータを使用しての作業は効率が良く，作業期間の短縮や省人化，さらには作業や部品の標準化など大きなメリットをもたらしてくれるからです．

その第一歩はCAD(Computer Aided Design：コンピュータによる設計支援)すなわち回路図の作図システムでした．

初期の頃のCADは正直なところ価格のわりにはそれほど使い勝手が良いとは言えませんでしたが，コンピュータの性能アップとGUI(グラフィカル・ユーザ・インターフェース)の向上により，格段と便利なものになりました．

それから，回路動作を解析する回路シミュレータとプリント基板設計用PCB CADが使われるようになりました．

さらには，プリント基板製造のためのCAM(Computer Aided Manufacturing：コンピュータによる製造支援)などコンピュータによる支援システムが続々と現れました．

ところが，これらの作業は回路設計/製作を行う一連の流れの中で行われるにもかかわらず，コンピュータ・システムでは各作業で閉ざされているという問題が生じてきました．

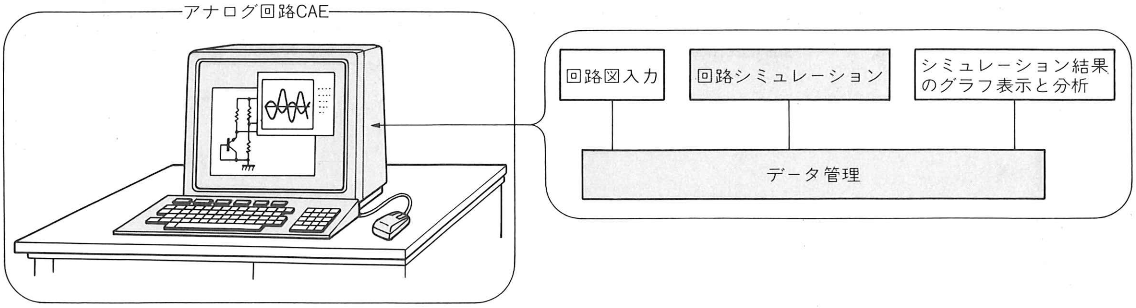
たとえば回路図の作成という作業を最初のラフスケッチのとき，あるいは回路解析のとき，さらにはプリント基板設計のときにそれぞれ行う必要があったのです．

とりわけ回路設計(回路図作成・評価・部品決定)は一人の技術者が連続した作業の中で行うわけですからなおさらのことです．また，GUIもシステムにより異なるので，それぞれの使用方法を会得する必要も出てきてしまいます．

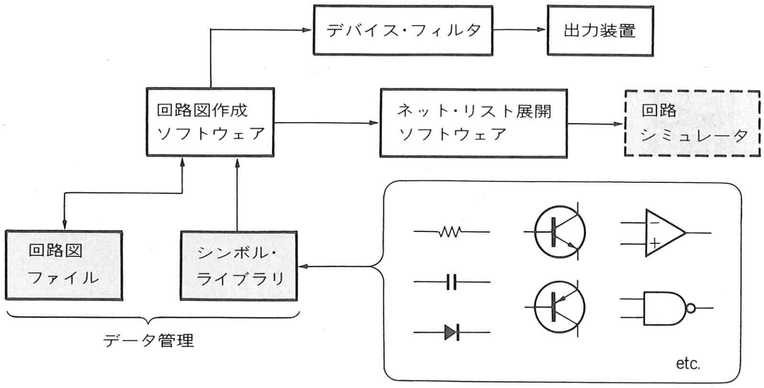
そこで，これらの一連の作業を効率良く行うためのツールとして図1に示す中でもっとも広い範囲をカバーするCAE(Computer Aided Engineering：コンピュータによる技術開発支援)が登場することになります．

CAEは広範囲で，しかも回路動作の評価という回路設計の中でもっとも重要な作業を支援します．

〈図2〉 CAEの概念



〈図3〉 回路図入力の概念



そこで，回路CAEのおおよその概念を示したものが図2です．これらの四つの機能がすべて整備されていて，はじめて回路CAEシステムとなります．特に回路動作の確認を行う回路シミュレータがなければCAEとは言えません．

また，この中でデータ管理の部分は，さらにいくつもの要素に分かれています．このため，なかなか理解しにくい部分です．しかしながら，ユーザにとってはこの部分が非常に重要な役割りをもっています．この部分の機能を良く理解していないとCAEを導入しても使えないことにもなりかねません．

以下，順にCAEの機能を解説することにします．

回路図入力とデータ管理

回路設計はまずラフな回路作成から始まり，次に回路シミュレーションを何度か行うことによって，徐々に完成度の高い(回路仕様に合った)ものに作り上げていきます．

そのような訳で，回路製作の最初は回路図入力の作業から始まります．その回路図入力の概念図が図3です．

ここで行うデータ管理の一つは回路を構成する素子を表す絵柄(シンボル)です．抵抗，コンデンサなど使用するシンボルは，すべてライブラリに登録されていなければなりません．というのも回路図作成ソフトで回路図を描く場合，このライブラリからシンボルを取り出して描くからです．

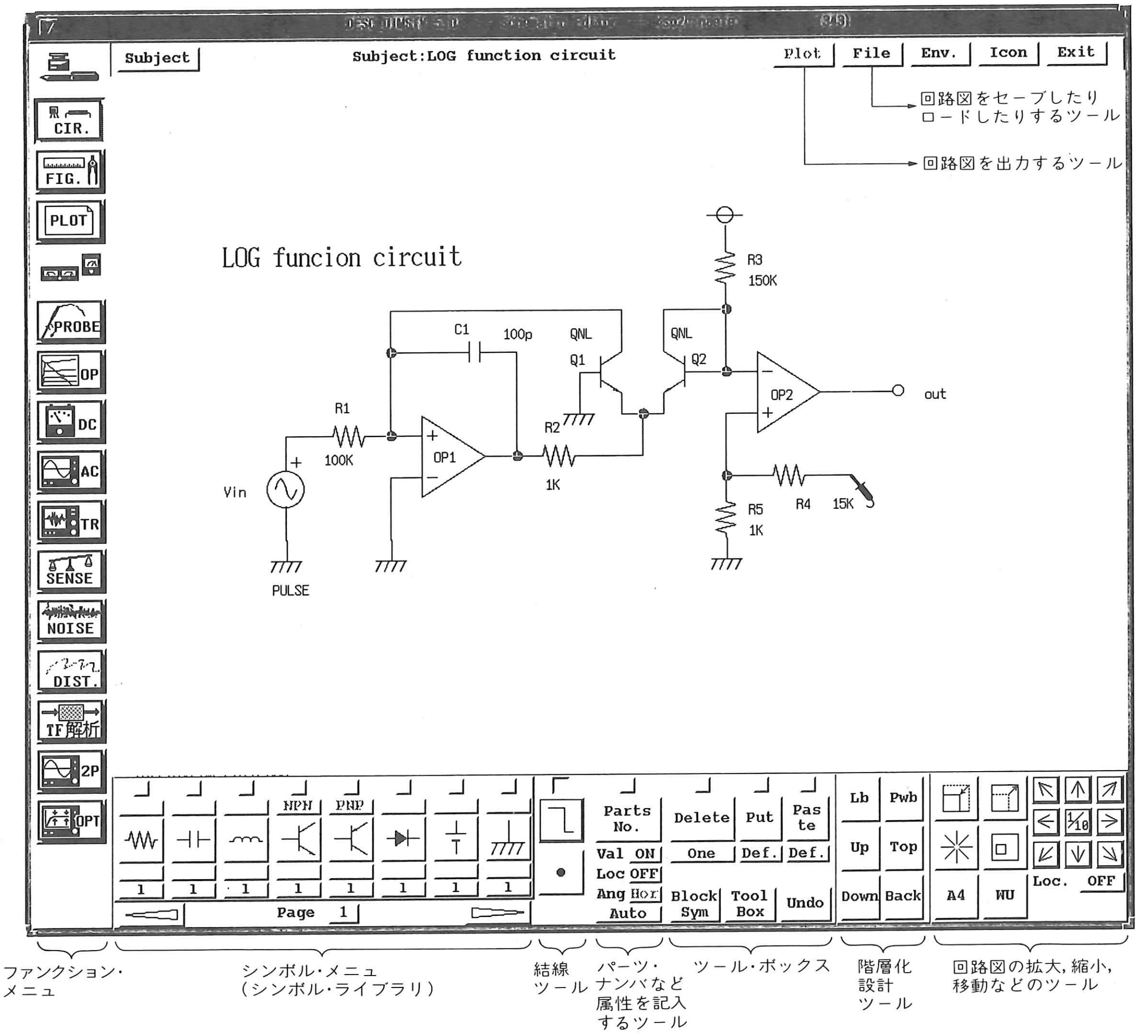
したがって，当然のことですがライブラリにないシンボルは描けないということになります．また，今は必要なシンボルがそろっていても将来追加したくなる物がでてきたり，あるいは変更したくなるような場合もあるでしょう．

このため，回路図入力にはシンボルを追加したり，あるいは消去したり，変更したりできる保守機能があるはずです(図3では省略)．

もう一つのデータ管理機能は，回路図のファイリング機能です．この機能は，作成した回路図をセーブしたり，ロードしたりする機能です．設計者自身あるいは他の設計者が既存の回路を参照したい場合があります．このために，使いたいときに即座に取り出せるように整理して保存しておく必要があります．

また，新規に回路設計を行う場合，過去に設計した実績のある回路の一部分を使用したい場合，さらには回路図をドキュメントの内部など，回路図以外のものに組み込みたい場合もあると思います．そのため，ここでは回路図のカット&ペーストの機能も必要となります．

〈図４〉 回路図入力の実際



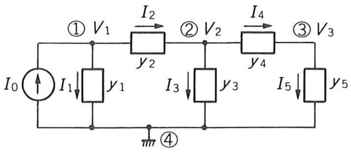
〈図５〉 回路図のネット・リスト

```
Vcc 14 0 5
R5 3 0 1K
Vin 5 0 PULSE(0, 3.5, 0, 5M, 0, 5M, 10M)
Q2 8 8 9 Pa70
R2 9 10 1K
C1 13 10 100P   →C1(100P)をノードNo.13と10につなぐ
R3 14 8 150K
R1 13 5 100K    →R1(100K)をノードNo.13と５につなぐ
Q1 13 0 9 Pa70
R4 12 3 15K
X19 13 0 10 Sa70
X20 3 8 12 Sa70
```

```
.SUBCKT Sa70 1 2 4
Rdi 1 2 500E6
Rout 0 4 1
Gopen 4 0  2 1  10000  IC=0
.ENDS Sa70
```
↑ OPアンプ（OP1，OP2）のサブサーキットによる回路モデル

```
.MODEL Pa70 NPN(
+ IS=1E-16
+ BF=80
+ NF=1
   ⋮ 略
+ AF=1
+ FC=0.5
+ )
.END
```
↑ トランジスタ（Q1，Q2）のパラメータ

〈図6〉 節点解析法



(a) 回路例

	ノード・アドミタンス・マトリクス(Y_n)			ノード電圧ベクトル(V_n)	電流源ベクトル(I_s)
	ノード①	ノード②	ノード③		
ノード①	y_1+y_2	$-y_2$	0	V_1	I_0
ノード②	$-y_2$	$y_2+y_3+y_4$	$-y_4$	V_2	0
ノード③	0	$-y_4$	y_4+y_5	V_3	0

$$\begin{bmatrix} y_1+y_2 & -y_2 & 0 \\ -y_2 & y_2+y_3+y_4 & -y_4 \\ 0 & -y_4 & y_4+y_5 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} I_0 \\ 0 \\ 0 \end{bmatrix} \quad \cdots\cdots\cdots\cdots(1)$$

(b) 回路方程式の作成方法

$$\left.\begin{aligned} I_1 &= y_1 V_1 \\ I_2 &= y_2 (V_1 - V_2) \\ I_3 &= y_3 V_2 \\ I_4 &= y_4 (V_2 - V_3) \\ I_5 &= y_5 V_3 \end{aligned}\right\} \cdots(2)$$

節点①においてキルヒホッフの法則より

$$I_1 + I_2 = I_0$$

(2)式を代入

$$y_1 V_1 + y_2 (V_1 - V_2) = I_0$$

$$(y_1 + y_2) V_1 - y_2 V_2 + 0 V_3 = I_0$$

節点②においてキルヒホッフの法則より

$$-I_2 + I_3 + I_4 = 0$$

(2)式を代入

$$-y_2 (V_1 - V_2) + y_3 V_2 + y_4 (V_2 - V_3) = 0$$

$$-y_2 V_1 + (y_2 + y_3 + y_4) V_2 - y_4 V_3 = 0$$

節点③においてキルヒホッフの法則より

$$-I_4 + I_5 = 0$$

(2)式を代入

$$-y_4 (V_2 - V_3) + y_5 V_3 = 0$$

$$0 V_1 - y_4 V_2 + (y_4 + y_5) V_3 = 0$$

以上より

$$\begin{bmatrix} y_1+y_2 & -y_2 & 0 \\ -y_2 & y_2+y_3+y_4 & -y_4 \\ 0 & -y_4 & y_4+y_5 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} I_0 \\ 0 \\ 0 \end{bmatrix}$$

(1)式と同じになる.

(c) 回路方程式の検証

さらに，回路図をプリンタなどから出力することもあります．このため，デバイス・フィルタ，すなわち使用したいプリンタやX-Yプロッタへのデータ変換ソフトウェアが整っていなければなりません．

そこで，図4に回路図入力の例を示します．これは回路CAEの多くの機能から，左にあるファンクション・メニュから回路図入力を選択し，図示したものです．

前述のように素子のシンボルはライブラリに登録されており，シンボル・メニュから自由に取り出すことができます．作図は，メニュからマウスで指定して作図エリアに配置することで行います．

また，シンボルは数が多いため何ページにもまたがっており，これらのシンボルはユーザが簡単に再配置できるようなしくみになっています．

さて，図3に戻りますが，作成した回路図を回路シミュレータで解析する場合，ネット・リストに展開する必要があります．この場合のネット・リストとは回路シミュレータが理解できる回路の表現方法です．

図4の回路図のままでは単なるシンボルと線の集まり(グラフィックのデータ)にすぎません．回路シミュレータにはどんな素子がどことどこにつながっているかというデータが必要なのであり，グラフィック・データは必要ありません．そこで，素子と素子との接続のみを表すネット・リストとして必要なデータのみが回路シミュレータに送られます．

図5に図4の回路図(対数アンプ)のネット・リストを示します．図中のノード番号は回路図入力のソフトウェアが自動的に決めた番号で，これにより各素子の接続情報と素子の特性がわかります．

リスト中の回路モデルとパラメータについては次項で詳しく解説します．

回路シミュレータ

回路モデルとかパラメータを理解するためにはコンピュータによる回路シミュレーションの仕組みを把握する必要があります．

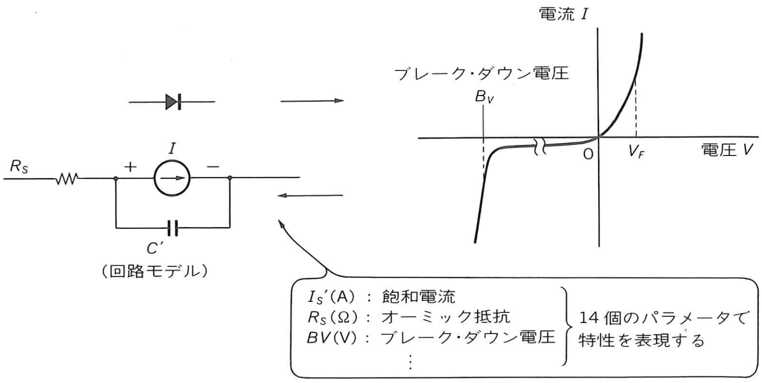
このため「急がば廻れ」のことわざにしたがい，まずは回路解析法について触れておきます．

● 回路解析法(節点解析法)

コンピュータで回路シミュレーションを行うためには，最初にメモリ上に解析する回路の回路方程式を作成しなければなりません．そして，回路方程式を解き，調べたいノードの変数(電圧，電流，位相など)をみることになります．

回路方程式の組み立て方法には，閉路解析法や状態変数解析法といったいろいろな方法があります．これらの中で，現在主流となっているのが修正節点解析法

〈図7〉ダイオードのモデル



です．

この修正節点解析法の源流である節点解析法を紹介します．

節点解析法は回路のノードの中から一つを基準節点として，その基準節点に対するすべてのノードの電位(ノード電圧)を回路方程式の未知変数にする方法です．この回路方程式は次式で表わされます．

$Y_n \cdot V_n = I_s$

ただし，

Y_n：ノード・アドミタンス・マトリクス

V_n：ノード電圧ベクトル

I_s：電流源ベクトル

図6(a)の回路例の場合，たとえばノード④を基準節点に選び電位を0と考えます．この時，ノード①～③のノード電圧をそれぞれ未知変数 V_1～V_3 とします．

回路方程式の作成方法は，まず図6(b)のように Y_n，V_n，I_s をテーブルに格納します．具体的にどのように格納するかと言うと，二つのノードを行，列番号または列，行番号とみて，その位置のところにノード間のアドミタンスをマイナスにして格納します．

ただし，その行，列番号間に何も接続されていない場合は0を格納します．なお行，列番号が同じ所にはそのノード番号に接続されているアドミタンスの和を格納します．

たとえば，ノード①の行に対してノード列①は，ノード①に接続されているアドミタンスの和なので，y_1+y_2 となります．また，ノード列②はノード①とノード②に接続されているアドミタンスのマイナスになるので，$-y_2$ となります．

ノード列③には何も接続されていないので，0となります．このようにして，ノード・アドミタンス・マトリクスを作成しますが，図6(b)からもわかるようにノード・アドミタンス・マトリクスは受動回路では対称行列になります．

次にノード電圧ですがノード①～③のノード電圧が，それぞれ V_1～V_3 なので，それぞれの未知変数をノード電圧ベクトルの項に格納します．

電流源ベクトルですが，ノード①に電流源 I_0 が接続されており，またノード②および③には電流源が接続されていないので，ノード①～③にそれぞれ I_0，0，0を格納します．

以上の操作により，図(b)の回路方程式のテーブルが完成します．そして，図中の(1)式の回路方程式が得られますが，図6(c)はこの方程式が正しいかどうかを検証したものです．

次に，(1)式の回路方程式から，コンピュータで未知数 V_1～V_3 を求めることにより回路の動作を調べることができます．

● 回路モデルとパラメータ

図6の回路のように，抵抗やコンデンサなどの受動素子のみで構成されている回路ならば，簡単にアドミタンス・マトリクスが求まります．したがって，回路方程式も比較的簡単に組み立てられると思います．

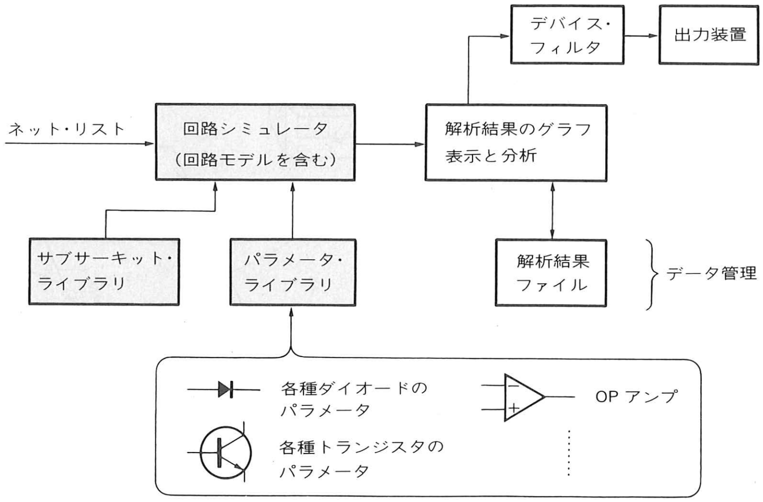
しかし，ダイオードやトランジスタなどの能動素子を含んでいたらどうなるでしょうか．アドミタンス・マトリクスは簡単には求まりません．ましてや，ICやLSIが含まれているとしたら，考えただけで混乱の極みへと行ってしまいそうです．

このための対策として，等価回路による素子のモデル化という方法があります．ダイオードやトランジスタなどについては具体的には受動素子や電流源などで等価な回路モデルを作成します．それができれば前述の方法でアドミタンス・マトリクスが求まり，回路方程式が立てられます．

さらに，ICやLSIは抵抗や電流源などに加え，モデルがすでにできあがっている小規模な素子なども含めて等価回路を作成します．そうすると回路モデル，さらには回路方程式も複雑にはなるものの，回路方程式はできるのでシミュレーションが可能になります．

かなりの大規模な回路になっても，ルールさえ決めてあげれば回路方程式の作成や方程式の解を求めるのもコンピュータは確実にこなしてくれます．したがっ

〈図 8〉 回路シミュレータの概念



て，回路シミュレーションにおいては，まず素子の回路モデルが正確でなければなりません．

以上が等価回路による回路モデルの作成方法ですが，これとは別にビヘービァ・モデルによる作成方法があります．

これは，素子のふるまいにのみ着目する方法です．確かにICなどの回路規模の大きいデバイスを等価回路モデルで表現すれば，正確にシミュレーションが可能です．

しかし，回路モデルの作成時間やシミュレーション効率が悪くなります．ビヘービァ・モデルに関しての説明は別にゆずることにし，ここではダイオードの等価回路モデルを考えてみます．

▶ ダイオードの等価回路モデル

まず，ダイオードの特性は図7のようになるので，基本的には図のような等価回路に置き換えることができます．I が正のときはPN接合順方向電流で，負のときはリーク電流を意味しています．また R_s はオーミック直列抵抗で，C は容量成分です．

これが一般的に使用されているダイオードの基本回路モデルです．しかし，ダイオードも基本動作は同じでも，すなわち回路モデルが同じでも用途により数多くの種類が存在し，フォワード電圧やブレーク・ダウン電圧などの特性はまちまちです．

このそれぞれの特性を表すのがパラメータです．すなわち，1S1588用のパラメータとかGP08用のパラメータと言うように各素子ごとにパラメータが存在します．

したがって，パラメータも回路モデル同様に正確さが要求されます．また，特にダイオードやトランジスタなど，半導体のパラメータについては他のものと区別して半導体パラメータと呼ぶこともあります．

回路モデルが完成し，パラメータもそろったら，あとは回路方程式をたてて解くだけです．回路シミュレータの概念図を図8に示します．回路シミュレータでは回路図入力から受け取ったネット・リストにしたがい，前述した方法で回路方程式を作ります．このため，ダイオードなどの回路モデルは通常シミュレータに組み込まれています．

そこで，その素子がどんな特性であるかを参照するためにパラメータ・ライブラリが用意されています．このライブラリにはシミュレーションするすべての素子のパラメータが登録されていなければなりません．

もっとも，デフォルトといって代表的な特性でシミュレーションする方法もありますが，この方法では回路の特性の大まかなふるまいは確認できても実際の部品を使用した場合の正確なシミュレーションはできません．

さて，図5のネット・リストを見るとトランジスタの半導体パラメータを取り込んだところが確認できます．

また，図5ではOPアンプ(ここでは理想OPアンプを使用)の回路モデルをサブサーキットで定義しています．

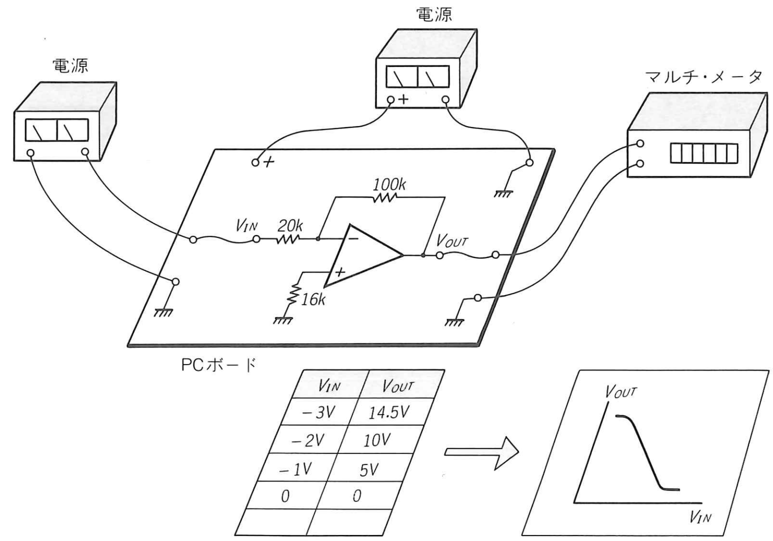
このように，ICなどをサブサーキットで登録することもよく行われます．また，サブサーキット・ライブラリではICなどの回路モデルのほか，よく使う回路を一つの部品として扱うこともできます．

そこで，頻繁に使う実績のある回路をサブサーキットに定義しておくことで同じような回路を作成する手間が省略できるなど，何かと便利なことがあります．

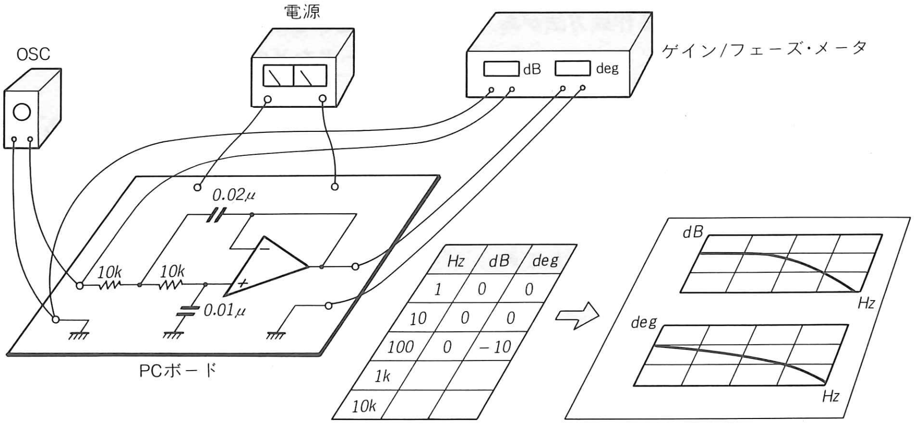
ご承知のように，ライブラリとはもともと図書館という意味ですが，必要なときに即座に本が取り出せるのと同じように，ユーザ自身がきちんと各種のデータやファイルを整理しておくことが必要です．

通常，回路CAEにはシンボル・ライブラリの保守機

〈図9〉 DC 解析のイメージ



〈図10〉 AC 解析のイメージ



能と同様に前述したライブラリの保守機能があります．特に半導体パラメータに関しては，h_{fe} などを変えてシミュレーションしたくなります．このため，保守機能が自由自在に使いこなせないようでは不便です．

図8に戻りますが，解析結果のグラフ表示・分析ソフトが設けられています．これは，シミュレーションした結果をわかりやすく表示させるソフトです．ここでは単に表示させるのみでなく，演算機能や統計データなどの分析・評価までできる機能が望まれます．

また，解析結果のファイリング機能はあまり重要でないように思われがちです．なぜかというと，再度シミュレーションすれば同じ結果が得られるからです．しかし，回路によっては何時間とかかる解析もありますから，このような解析結果はきちんと整理して保存しておくとよいでしょう．

● 解析機能

ユーザにとってもっとも重要な関心事は最終的には回路シミュレータでどんな解析ができるか，ではないでしょうか．そこで解析機能について述べたいと思います．しかし，一口に解析機能と言っても数多くの種類があります．そして，それもある CAE ではできるが他の CAE ではできないということもあり得ます．また，その逆のケースも起こり得ます．

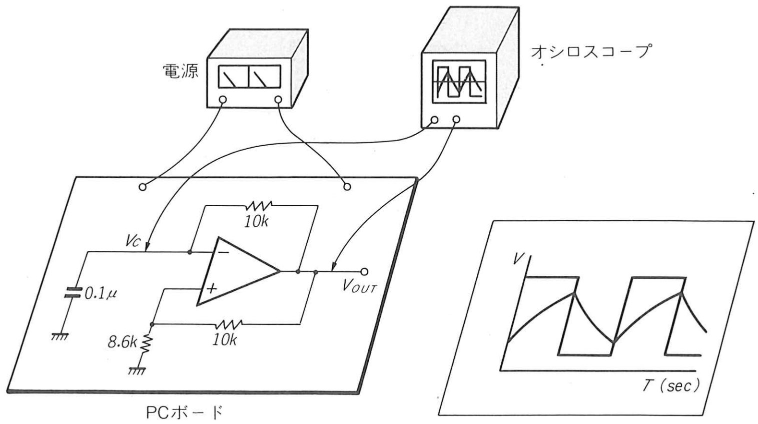
そんなわけで，ここではほとんどの回路シミュレータがもつ，基本的な解析機能を紹介するにとどめます．

▶ DC 解析（直流解析）

DC 解析は回路に直流電源を加え，入力電圧や温度，抵抗，半導体パラメータなどの値をスイープさせたときの直流出力応答（電圧，電流）を調べるものです．

測定器にたとえれば，マルチ・メータ，電圧計，電流計などがこの解析機能に相当します．たとえば，図9

〈図11〉 TR 解析のイメージ



はOPアンプによる反転増幅回路の入力電圧に対する出力電圧特性を測定する実験例です．

このような場合，実験では入力に直流電源を接続し，入力電圧を変化させて出力電圧を測定します．

しかし，回路シミュレータでは入力電圧をスイーパとして選択し，入力電圧の上限と下限，解析ステップを設定してDC解析を実行するとグラフおよび表として入力電圧-出力電圧特性を求めることができます．

また，温度をスイーパとして温度特性を求めたい場合もあります．このような場合には当然，各素子の温度特性が必要です．通常，CAEベンダから提供されるOPアンプなどの半導体パラメータには温度パラメータが含まれています．

しかし，抵抗などユーザが設定する素子はユーザ自身で入力しなければなりません．このデータを入れ忘れると実際の回路特性とは差異が生じてしまいます．

▶ AC解析（周波数解析）

AC解析は入力に正弦波を加え，その周波数をスイープさせたときの出力応答を求めるものです．

測定器にたとえれば，ゲイン・フェーズ・メータやネットワーク・アナライザに相当します．図10は，VCVS型2次LPFの周波数特性を求める実験例です．実験ではオシレータで入力の周波数を変化させ，その時の位相と利得をゲイン・フェーズ・メータで測定しますが，なかなか手間のかかる作業です．

しかも周波数が高くなると安定した測定が難しくなります．しかし，シミュレーションでは入力信号の周波数の上限と下限，解析ステップを設定し，AC解析を実行すると簡単に周波数特性が求まります．

また，詳しく結果を見たい場合には数値表を見ることもできます．

▶ TR解析（過渡解析）

TR解析は時間の経過によって変化する回路の動作を解析するときに使用します．オシロスコープがこの解析に相当する測定器です．

図11の実験例は非安定マルチバイブレータの周期や振幅をオシロスコープで求めるものです．しかし，オシロスコープの操作もなかなかわずらわしいうえ，画面の波形から周期や振幅を数値で求めるのも厄介なものです．

シミュレーションでは解析する時間とステップを設定してTR解析を実行すると簡単に求まります．

また，詳しく数値で解析したいときには，DC解析やAC解析と同様に数値表を見ることもできます．

▶ DFT解析

上記のTR解析では時間領域における特性を調べることができます．DFT解析はこの時間領域の特性から，どんな周波数成分をもっているかを解析します．

すなわちDFT解析は時間領域で得られた波形を任意の区間においての周波数スペクトルに分解するわけです．測定器にたとえれば，スペクトル・アナライザやFFTアナライザがこの機能に相当します．

回路CAEではこのような測定器の機能が組み込まれていて，しかも測定器を操作するよりも簡単な操作で使用できます．DFT解析を行いたい場合は時間領域で得られた波形に対し，周波数分析をしたい領域を指定してDFT解析を実行するだけです．

ただし，この機能は本来回路シミュレータが有する機能ではなく，図8中の分析ソフトがもつ機能です．このようなことを考えると分析ソフトの機能も重要です．

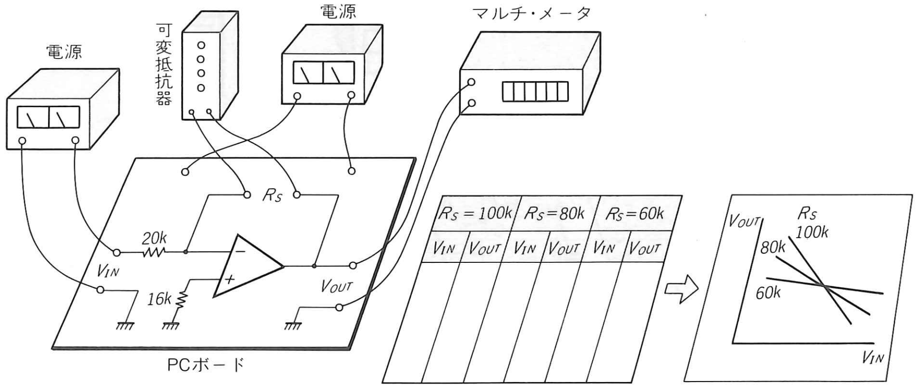
▶ パラメトリック解析

パラメトリック解析は，電源や温度，抵抗，コンデンサ，半導体パラメータなどの値をパラメータとして変化させて解析を行い，結果を1枚のグラフに重ねて表示させる解析です．

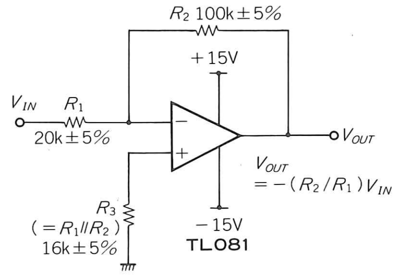
前述のDC解析，AC解析，TR解析の三つの基本解析に適用できます．

回路定数を決める場合，実験でも部品を何度か変えてデータを取り検討します．図12は反転増幅回路の抵

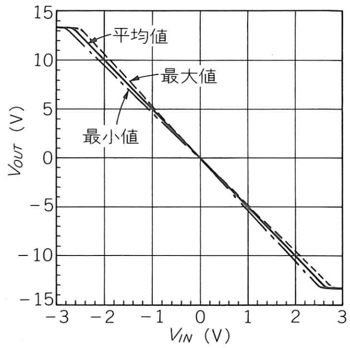
〈図12〉 パラメトリック解析のイメージ



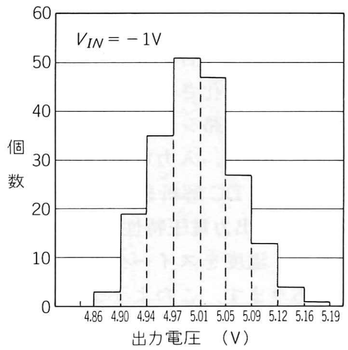
〈図13〉 例にとる反転増幅回路



〈図14〉 モンテカルロ解析の結果



〈図15〉 ヒストグラム表示



抗を変化させて入力電圧-出力電圧特性を測定している例です.

まず R_s を 100 kΩ として入力電圧を変化させて出力電圧を測定します. そして R_s を変化させ, 同様の実験を繰り返します.

シミュレーションではパラメータとする素子と定数を設定し, この場合にはDC解析を実行することで一度に結果が出力されます.

▶ モンテカルロ解析(統計解析)

一般に, 回路部品の特性は一様ではなく, ばらつきがあります. このような部品を用いて回路を量産すると回路の特性にもばらつきが発生します.

モンテカルロ解析は回路を量産した場合の特性のばらつき状態を予測するのに役立ちます.

そして, 量産した場合のばらつきを少なくするために必要以上にばらつきの少ない高価な部品を使用して製品原価を高くしてしまうようなことを防ぐためにも便利な解析機能です.

すなわち, ばらつきのある部品すべてをそのばらつきの範囲内でランダムに値を設定してシミュレーションするわけです. そして, これを多数回, たとえば100回とか200回実行し, 全体の回路特性の最大, 最小, 平均などを調べます. さらにヒストグラムなどにより, くわしく特性のばらつきを見ることもできます.

例として, 図13の反転増幅回路のDC特性のモンテカルロ解析の結果が図14ですが, 特性のばらつきがわかります.

ここでは200回実行した結果の最大, 最小, 平均のみをプロットしました. V_{IN} が−1Vのときのヒストグラム(X軸:出力電圧, Y軸:個数)が図15です.

以上は, モンテカルロ解析をDC解析に適用した例ですが, AC解析やTR解析についても同様にモンテカルロ解析が適用できます.

▶ 感度解析/ワースト・ケース解析

前項は反転増幅回路の抵抗に5%のばらつきのあるものを使用した場合の出力電圧のばらつきをモンテカルロ解析で調べた結果です.

そこで, さらにばらつきを少なくするためにはどうしたらよいかを考えてみます. 答えを言うとこんなとき使用するのが感度解析/ワースト・ケース解析です.

シミュレーションの結果が図16ですが, それぞれの素子の中心値, 最大値, 最小値が出力されます. そして, それぞれの素子の素子感度とノーマライズ感度が出力されます.

素子感度とは素子値が単位量変化したときの出力の変化量です. たとえば抵抗では, 1Ω変化したときの

〈図16〉感度解析/ワーストケース解析の結果

```
==============================================================================
Sensitivity Analysis     (最小値)              (素子感度)    (感度レベル)
output = Vout
element        nom value  min value  max value   ele sens   nor sens sens level
         (中心値)              (最大値)       (ノーマライズ感度)
R1 (5%)            20000      19000      21000  0.0002512    0.05025 ********************
R3 (5%)            16000      15200      16800  8.851e-12  1.416e-09


R2 (5%)           100000     105000      95000 -5.025e-05   -0.05025 ********************
------------------------------------------------------------------------------
Worst Case Analysis   (Vin=1.0V)
      nominal = -5.029939e+00 ←── 計算上の出力の中心値
      minimum = -5.558873e+00 ←── 計算上の出力の最小値
      maximum = -4.551379e+00 ←── 計算上の出力の最大値
==============================================================================
```

出力の変化量のことです．図16の場合，R_1 と R_3 は抵抗値が＋に変化すると出力も＋に変化し，R_2 は抵抗値が＋に変化すると出力が－に変化しています．

ノーマライズ感度とは素子値の中心値の１％あたりの感度で，この値のほうが素子感度よりも実用的な値です．そこで感度レベルでは各素子のノーマライズ感度をバーグラフで比較できるようになっています．

すなわち，図13の反転増幅回路では R_1 と R_2 のばらつきの影響を一番大きく（ばらつき方向は反対方向）受け，R_3 のばらつきの影響はほとんど受けないことがわかります．

したがって，R_1 と R_2 の抵抗を精度の良いものにすることで回路のばらつきを少なくすることができます．

ワースト・ケース解析とは素子値のばらつきから考えられる計算上の中心値，最小値，最大値を求める解析です．ここでは５％の抵抗を使用した場合，$V_{in}=1$ V のとき，最小値が－5.559 V で最大値が－4.551 V になります．

もちろん，この範囲が狭いほど部品のばらつきによる回路のばらつきが少なくなります．そこで，各素子のばらつきを変化させて感度解析/ワースト・ケース解析を実行し，ワースト範囲を狭めていきます．

CAE 導入のポイント

回路シミュレーションの利点を理解できても，いざ回路 CAE を導入しようとするとどれが良いか迷うものです．もちろん，費用や用途などにより検討方法も異なります．そこで，ここでは導入の際の共通した注意点やチェック・ポイントをまとめてみました．

● 回路モデルとパラメータ

前述したように回路モデルがなければ解析は不可能です．

このため，使用する部品の回路モデルが整備されているかどうかを確認する必要があります．

また，使用していると自分で回路モデルを作成したくなる場合もあります．そのようなわけでこのためのツールも必要です．もし，使いやすいツールと使いこなせる技術があればモデルがサポートされていない部品があっても解析は可能です．

パラメータについてもいくつか確認する項目があります．当然ですが，使用する素子のパラメータがなければなりません．そして，重要なことはそのパラメータの精度です．数はそろっているが簡易的に作成されているために精度が悪いといったものもあります．

OP アンプなどの IC はカタログから予測できますが，ダイオードやトランジスタや FET などの素子は実際に測定したものがベストです．温度に関するパラメータも同様です．

また，半導体パラメータは DC パラメータと AC パラメータがあります．とくに AC パラメータは測定が難しくサポートされていない場合もあります．高周波などでは AC パラメータがサポートされていないと解析結果が大きく異なってしまう場合があります．

● 収束性と解析速度

モデルやパラメータが整備されていても答（解析結果）が正確に出なければ意味がありません．

解析速度が速くて収束性が良いのがベストですが，一般に解析速度を速くすると収束性や解析精度が悪くなり，逆に収束性や解析精度を求めると解析速度が遅くなるという相反関係にあります（図17）．

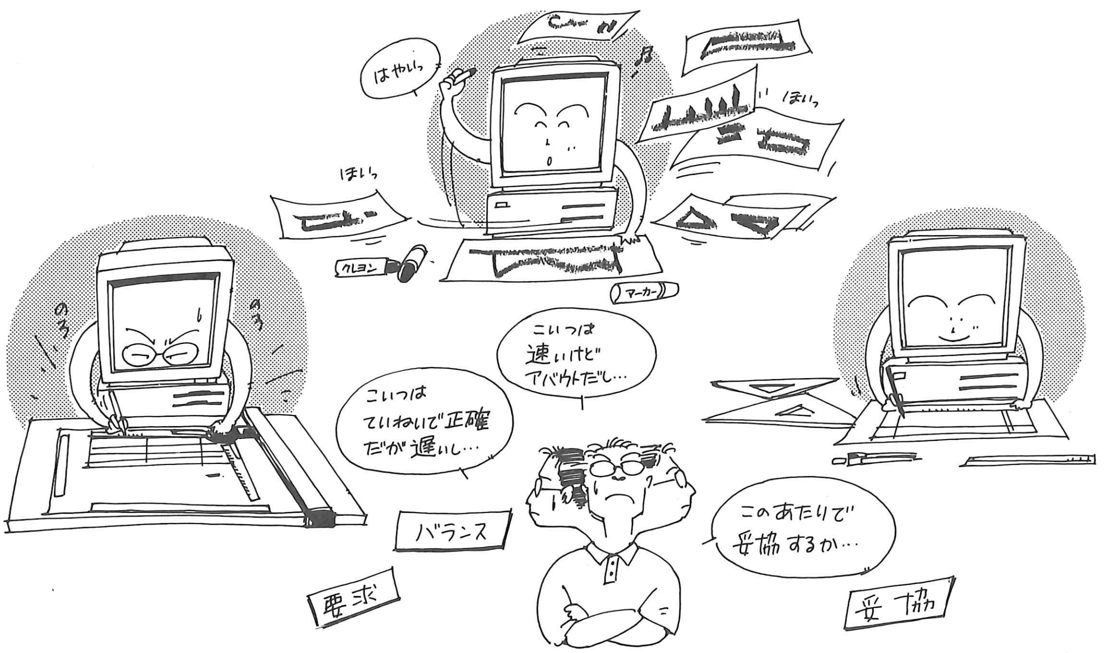
このため，シミュレータではいくつかのオプションを設定してほどほど満足のいくところで使用します．このオプションの項目はシミュレータによりまちまちで，しかもオプションのデフォルト値（ユーザが特に設定しないときの値）も違っています．

そのような理由から，あるシミュレータでは動作するが他のシミュレータでは動作しないという（シミュレータにとって）特殊な回路もあります．

しかし，このような場合，ほとんどが何がしかのオプションを設定すれば解析速度の差は生じますが動作します（筆者の経験ですが）．

そのようなわけで，導入の前にはいくつかの回路でベンチマーク・テストを行っておいたほうが無難です．また，CAE ベンダにオプション設定方法などの技術サポートができることも確認しておくことです．

〈図17〉 収束性と解析速度は相反する関係



● **その他**

上記2点がもっとも重要と思われるものですが，その他いくつか思い当たるチェック・ポイントを述べます(図18).

先に代表的な解析機能をあげましたが，そもそもCAEが使いたい解析機能をサポートしているかどうか，これなどは最重要項目です．また，ソフトは次第に進化していくので，現在サポートされていない機能でもユーザの意見を取り入れてバージョンアップしてくれるソフトには魅力があります．

エラー・メッセージが的確に出るかどうかも注目したい項目です．

CAEを使用していると回路図の描き間違いや定数の抜けなど，思わぬミスを起こすこともあります．そんな簡単なミスのために多くの貴重な時間を費やしてしまいかねません．このとき，的確なエラー・メッセージを出してくれればこれが大きな助けとなります．

また，使用しているうちに使い方がわからなくなる場合もあります．回路設計ばかりでなく，他の仕事もする人はなおさらです．このような場合，助けてくれるのは日本語のオンライン・マニュアルです．

外国製のソフトを購入すると，よく英文のマニュアルがどさっときますが，これを参照しながら使うのでは効率が悪すぎます．使い方がわからないところで必要な項目だけを教えてくれるオンライン・マニュアルは貴重な戦力です．

最後になってしまいましたが，操作性もたいへん重要なチェック・ポイントです．いくら機能が良くても操作性が悪ければ使いこなすことはできません．

しかし，この操作性はどうも個人差があるように思われます．使いはじめの人や長年使用している人，あるいは扱う回路によっても異なるように思われます．

したがって，導入の際には実際に何度か試用して確認してみることが良い方法と思われます．

操作環境

回路CAEを動作させる環境は大きくパソコンとUNIXワークステーションに分けられます(図19).

● **パソコン**

最終的には図3の回路図入力と図8の回路シミュレータを使ってCAEの環境を構築することになります．

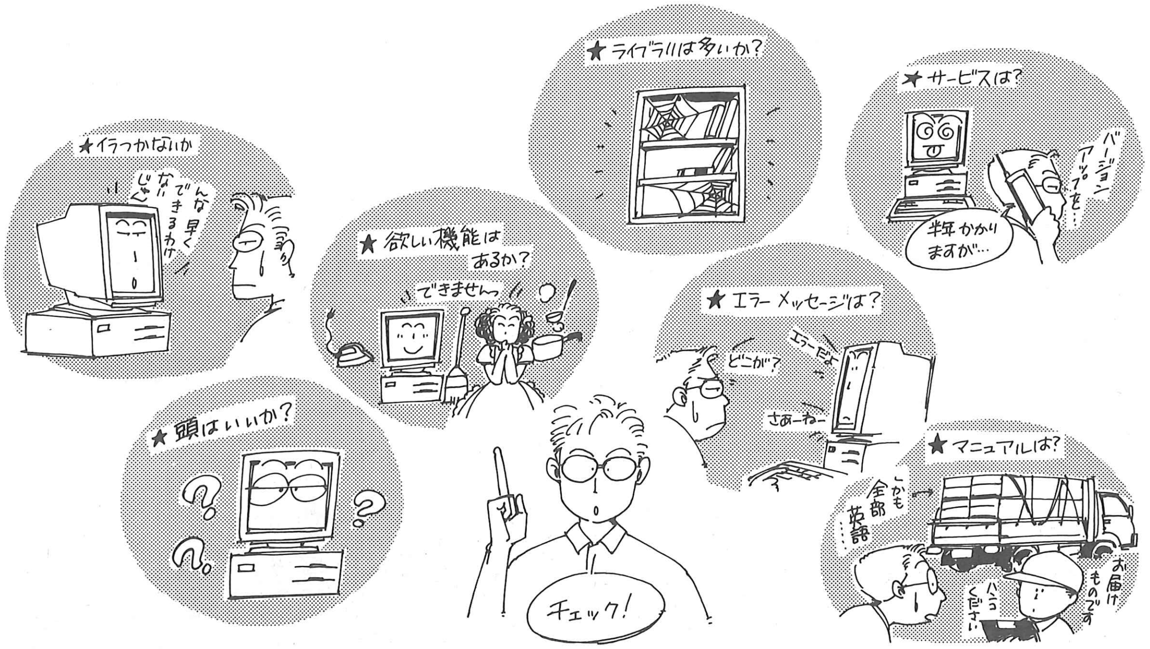
図3の回路図入力の目的は回路シミュレータが理解できるネット・リストを出力することです．したがって，エディタで直接図5のようなネット・リストを作成して回路シミュレータに入力する方法もあります．

比較的小規模の回路や回路設計の際，あまり大きな変更をしないとわかっている場合はこの方法でも良いでしょう．しかし，見慣れた回路図上で回路の検討ができないという欠点があります．

このため，やはり回路図入力は用意したほうが無難と考えます．回路図入力ソフトは本誌でも広告欄に掲載されていますし，評価版ソフトも頒布されています．

導入のチェック・ポイントでも書きましたが，失敗

〈図18〉 CAE を選ぶ際のチェック・ポイント



を防ぐためには実際にさわってみることが必要です.

このため，回路シミュレータ同様，一度評価版を購入し，十分確かめたうえでカスタマ版を購入することをおすすめします.

回路シミュレータはシミュレータ(結果の表示は含む)とライブラリを分けて販売されるのが普通のようです．チェック・ポイントを思い出してよく検討して導入してください.

一般的には回路図入力と回路シミュレータは同じメーカのものを購入したほうがデータ・フォーマットや将来のバージョン・アップなどのことを考慮すると安全かと思います.

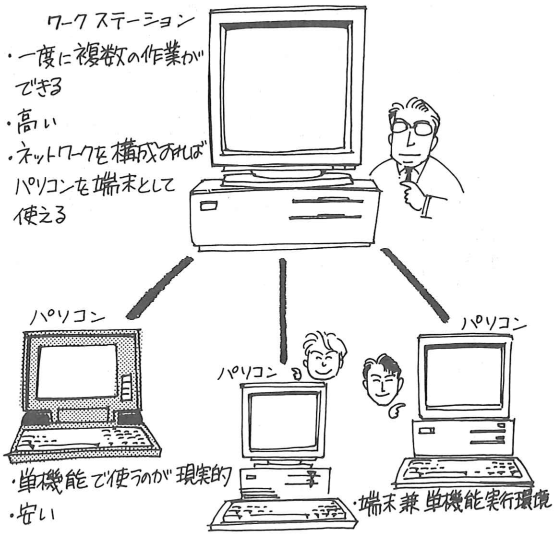
● UNIX ワークステーション

以前はEWSと言われていましたが,最近ではUNIXワークステーションという呼び方が一般的になりました．UNIX ワークステーションの場合には非常に単純です．と言うのも，1枚のウィンドウで図4のように回路図入力を，さらに他のウィンドウで各種のシミュレーションを起動できるからです．最近ではパソコンもWindowsが使われるようになり,また一部ではパソコンとワークステーションの差がなくなり，この例とまったくかわらない環境を実現できるものもあります.

図の例は左のオペレーション・メニュで回路図入力(左最上段のCIR)の環境にしたところです.

この後,たとえばDC解析を実行するのならば,メニュからDC解析を起動して(DCと書かれたメニュをマウスでクリック)解析を実行できます.

〈図19〉 パソコンと UNIX ワークステーション



したがって，ネット・リストなど煩わしいことはいっさい考える必要はありません．ライブラリなどもきちんと整備されています．また，インストールもベンダのSEが出向いて行ってくれます.

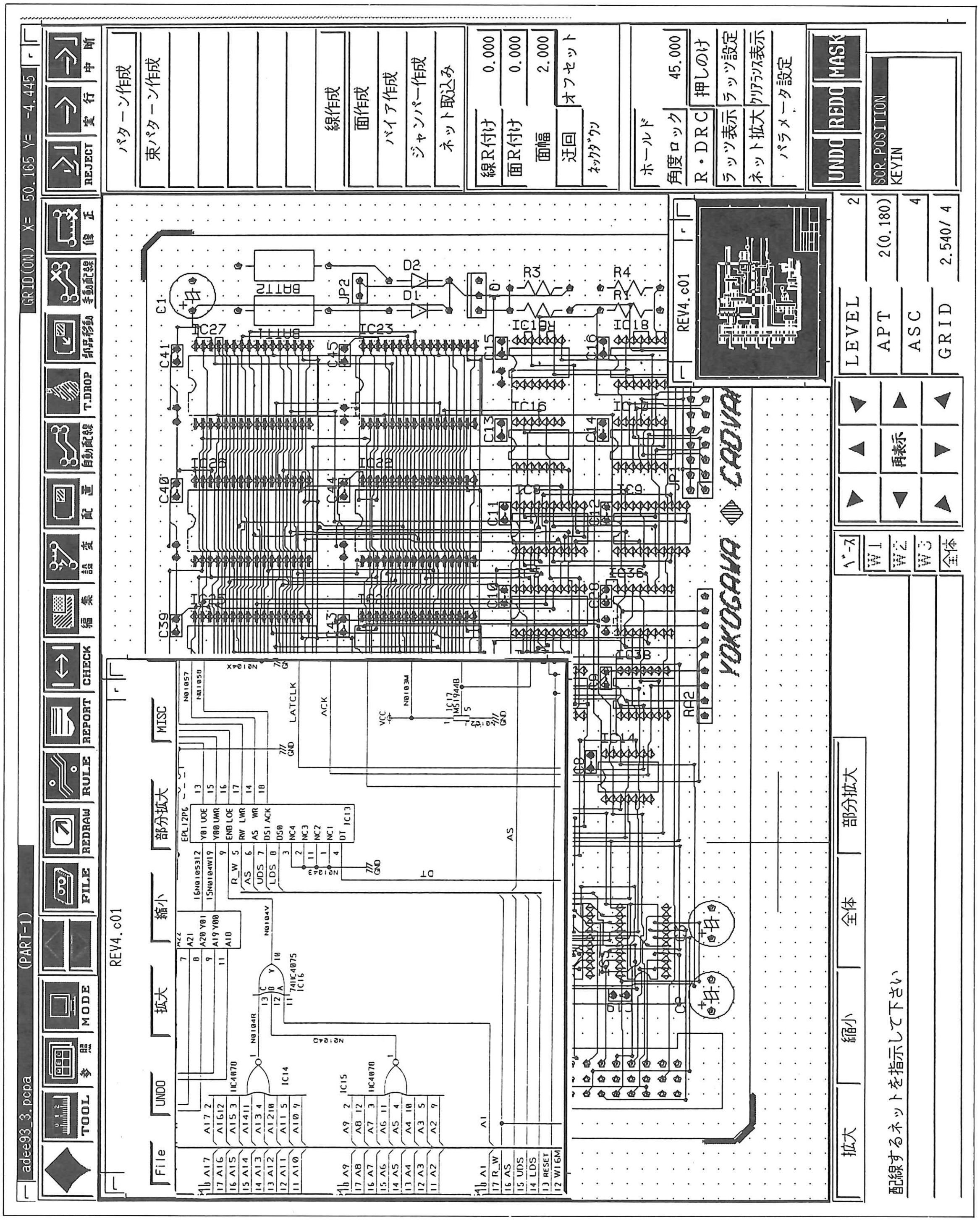
しかしながら，価格がパソコンに比べて高いというのが難点です．このため，ソフトがモジュールごとに分かれて販売されています．価格を抑えるためには必要なモジュールのみを購入し，我慢できる機能は極力導入を控える方法をお勧めします.

● パソコン版とワークステーション版の違い

非常に難しい選択ですが，回路図を描くのが主ならパソコン，業務用でシミュレーションすなわち回路CAEを主体として使用するならワークステーション(もしくはハイグレードなパソコン)を選ぶべきでしょう．

回路図の作成CADのように導入後のメンテナンスが少なく，CPUやメモリの負荷も少ないものはパソコンでも問題はないと考えます．値段を考えればパソコン

〈図21〉 PCB設計までできるCAE



のほうが有利です．

しかし，導入後のフォローや重い計算を必要とするCAEではやはりワークステーションが必要になります．特に統計解析や複数のシミュレーションとなると多くのパソコンでは現実的に難しいと言わざるをえません．

複数のユーザでLANなどによってソフトやライブラリを共有するようになればなおさらです．また，ワークステーション自体，最近では安くなってきているうえ，ソフトのほうもそれにともなっても安くなってきています．

しかし，パソコンの性能向上やソフトの進歩もめざましいものがあることは否定できません．このため，比較的小規模な回路で，開発効率を問わなければ前述のように個人用回路CAEとして使用できるでしょう．

このへんの見極めは実際によく使う回路でベンチマーク・テスト（解析精度と処理速度などを評価）をして結論づけるべきでしょう．

いずれにしても価格ではパソコン，性能や解析速度やアフタサービスではワークステーションということになります．

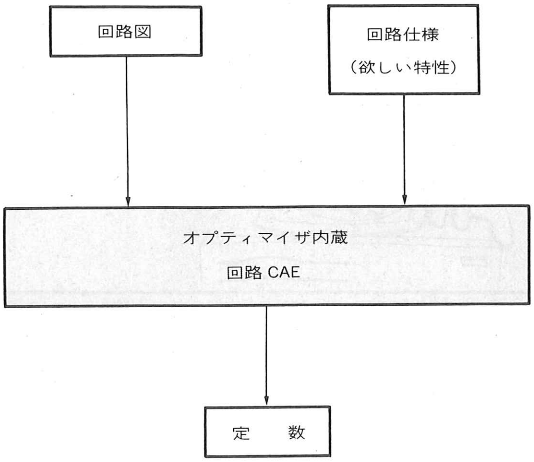
● 今後の動向

回路CAEの今後ですが，二つの大きな動きがあります．

その一つはオプティマイザの出現です．

従来の回路CAEは回路図に素子データ（パラメータや定数）を加えシミュレーションを行い，結果が望みの特性になるまで繰り返し行っていました．確かに実験の代わりにシミュレーションをしているという見かたをすれば効率は上がりました．しかし，何度もシミュレーションを行う必要があるので解析をくり返し行っていること自体，まったく変わりがありません．

〈図20〉オプティマイザの動作



そこで，オプティマイザを使用することによって，回路図と回路仕様（求める特性）を入力して，回路仕様にもっとも近くなる素子データを一気に求めてしまおうというわけです（図20）．

現在のオプティマイザは，まだまだ改良の余地はありますが，これによりさらにシミュレーション効率が上がるとともに，熟練者でなくても回路設計が可能になります．

もう一つは，回路CAEとPCB CADの一貫システムです．回路CAEで回路設計をした後は，図1のように通常はプリント基板の設計を行います．そこで図21のように回路CAEで設計した回路図をPCB CADに入力して同じ環境でプリント基板の設計を行ってしまおうというわけです．

このように，コンピュータで回路設計から基板設計まで，いわゆる試作レスの時代がそこまで来ています．

次号予告

特集　CAD/CAE総ガイド（'93年11月号：10月9日発売）

回路設計にCAD/CAEを使用するのが当然の事になるのも，そう遠くはないでしょう．事実，CAD/CAEはものすごいいきおいで普及しています．

そんな現状を反映して，回路設計用/基板設計用，シミュレータ，オートルータなどなどパソコン用/ワークステーション用エンジニアリング・ツールが数多く発売されています．

今回の特集でも解説がありましたが，エンジニアリング・ツールの選択にあたっては，できれば本物，それができなければ評価版または試用版を実際に使ってみるのが，そのツールの使い勝手を理解するための早道です．

次回の特集は，その一つ前の段階として，どんなエンジニアリング・ツールが市場に出回っているのか，代表的なものを紹介します．これから導入を考えている読者への参考となることでしょう．



実用 ビヘービァ・モデル作成術①

3端子レギュレータとサーミスタ

苗手 英彦

回路CAEを使用して回路設計を行うと確かに効率良く設計ができます．そして，信頼性の高い設計ができることなど，多くの利点があることも承知のとおりです．

しかし，これらの利点は使用するデバイスの回路モデルやパラメータ・ライブラリが整備されていてはじめて実現できます．

このため，回路モデルができていない特殊なICや，モデルはあってもデバイス・パラメータのない素子などを使おうとするとけっこうやっかいなものです．

回路モデルが完成しているOPアンプなどでは，たとえデバイス・パラメータが整備されていなくてもカタログからデータを抽出したり，あるいはデフォルト値を使用してシミュレーションすることも可能です．

しかし，回路モデルがまだ整備されていない大規模なICを使用している回路をシミュレーションする場合，まず正しい回路モデルを作成しなければなりません．

この場合，等価回路をサブサーキットで定義する方法もありますが，この等価回路が入手できなければほとんど不可能です．

また，仮に入手できたとしても各素子の定数や半導体素子のパラメータを正確に把握することは意外に難しいものです．

このような場合，CAEの有効性は理解していても，CAEによる設計はあきらめざるを得なくなってしまいます．そこで，デバイスの特性をもとに，比較的簡単にデバイスの回路モデルを作成する方法を紹介します．

それがタイトルにあるビヘービァ・モデルです．

ビヘービァ・モデルとは

ビヘービァ(behavior)とは，もともと動作やふるまい，(機械などの)動きぐあいなどを意味する名詞です．

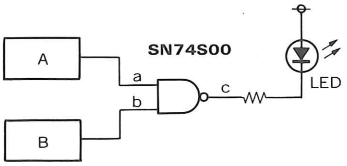
すなわち，ビヘービァ・モデルとは記述言語でデバイスの動き(特性)を表現した回路モデルと言えましょう．

たとえば，図1-1のLEDドライブ回路をシミュレーションする場合を考えてみます．

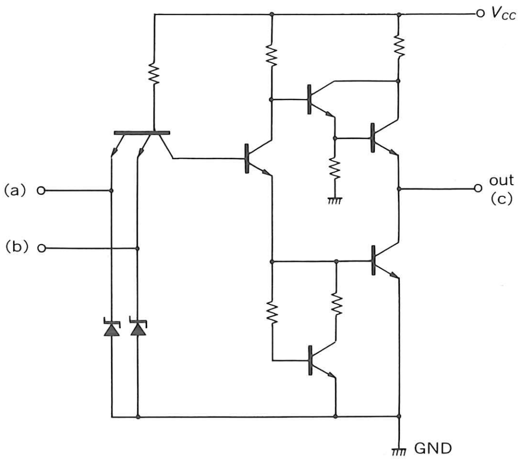
この回路は，回路Aと回路Bの出力aとbがともに"H"の場合にのみ，LEDを点灯させる回路です．この場合，CAEでシミュレーションするにはNANDゲートの回路モデルが必要です．

しかし，通常ではアナログ回路CAEにはディジタルICの回路モデルは登録されていません．そこで，ユー

〈図1-1〉 LEDドライブ回路



〈図1-2〉[1] SN74S00の内部回路



〈表1-1〉[1] SN74S00の主な特性

項 目	min.	typ.	max.
電源電圧	4.75 V	5 V	5.25 V
"H" レベル出力電流			−1 mA
"L" レベル出力電流			20 mA
"H" レベル入力電圧	2 V		
"L" レベル入力電圧			0.8 V
"H" レベル出力電圧	2.7 V	3.4 V	
"L" レベル出力電圧			0.5 V

ザが回路モデルを作成する必要があります．

このため，カタログからNANDゲートの内部回路(図1-2)を入手し，回路モデルを作成することになります．しかし，これだけの情報では，抵抗を何Ωにしたらよいのかわかりません．

また，半導体のデバイス・パラメータもどう設定したらよいかもわかりません．回路シミュレータを使ったことのある人でしたら，このような経験は一度はあると思います．

そこで，このNANDゲートの主な特性(表1-1)を考えてみます．もっとも重要な点は，電源を5V±5％に設定し，二つの入力電圧が2V(“H”レベルと認識される最低電圧値)以上になると出力は20mA以内で0.5V(出力が“L”レベル時の最大電圧)以下になるということです．

つまり，それ以外の条件では，出力は“H”レベル，すなわち少なくても2.7V以上になると解釈できます．

これを，条件文(if文)を使って表すと，

```
if a>2, b>2
then c<0.5
else c>2.7
```

となります．

すなわち，上記の条件文を回路モデルと考えても基本的な動作のシミュレーションが可能であるということです．

図1-2を回路モデルとした場合よりは正確さに欠けますが(実際のビヘービア・モデルではもっと正確なモデル化が可能)，シミュレーションの効率は優れています．

これがビヘービア・モデルの基本的な考え方です．

● ビヘービア・モデルの作成方法

ビヘービア・モデルはたいへん便利で，しかも有用なモデルですが，すべての回路シミュレータで使えるとは限りません．使えない回路シミュレータもあるので注意が必要です．

しかし，ほとんどの回路シミュレータでサポートされているので，マニュアルなどで確認してください．

さて，ビヘービア・モデルの作成方法ですが，残念なことに回路シミュレータによってそれぞれ作成方法が異なります．

したがって，Aという回路シミュレータで動作していたビヘービア・モデルがそのままBというシミュレータで動くとは限りません．しかし，記述方法は異なるものの，作成方法(設計の考え方)は同じです．

そこで本稿では，設計の考え方を中心に話を進めることにします．しかしながら，具体的な例(サブサーキット・ファイル)があったほうが理解しやすく，さらに読者の方が実際に作成する場合に，参考になると思います．

〈図1-3〉使用できる数式

```
a=b+c(加算)
a=b-c(減算)
a=b*c(乗算)
a=b/c(除算)
a=b∧c(AND)
a=f(b)(関数)
```

〈図1-4〉使用できる特殊関数

```
・sin( ), cos( ), tan( )
・hsin( ), hcos( ), htan( )
・asin( ), acos( ), atan( )
・ahsin( ), ahcos( ), ahtan( )
・exp( ), log( ), log10( )
・abs( )
・sqrt( )
```

〈図1-5〉条件文の書式

```
if(条件式)
  {変数a=○○○;変数b=×××;……;}
else
  {変数a=□□□;変数b=△△△;……;}
```

そこで具体例として，筆者が使用している回路CAEの例で示します．このCAEの回路シミュレータは一般的に広く使われているSPICE系の回路シミュレータなので多くの方の参考になるものと思われます．

さて，ビヘービア・モデルの作成は基本的にサブサーキット内で，「define block」内で定義することで行います．

すなわち，

```
.SUBCKT　サブサーキット名　ノード番号
  ⋮
define
  ⋮
  ビヘービア・モデルの記述
  ⋮
exit
  ⋮
.ENDS　サブサーキット名
```

というようにして使用します．

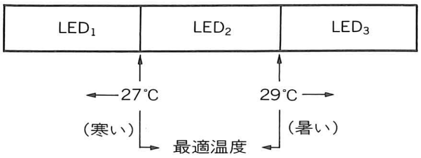
また，「define block」は，

```
define
function_name(v1, i1, v2…p1, p2…)
  ⋮
関数定義
  ⋮
function_name=…;
exit
```

のように定義します．また，モデル関数は，

・代入式，数式

〈図1-7〉 温度表示

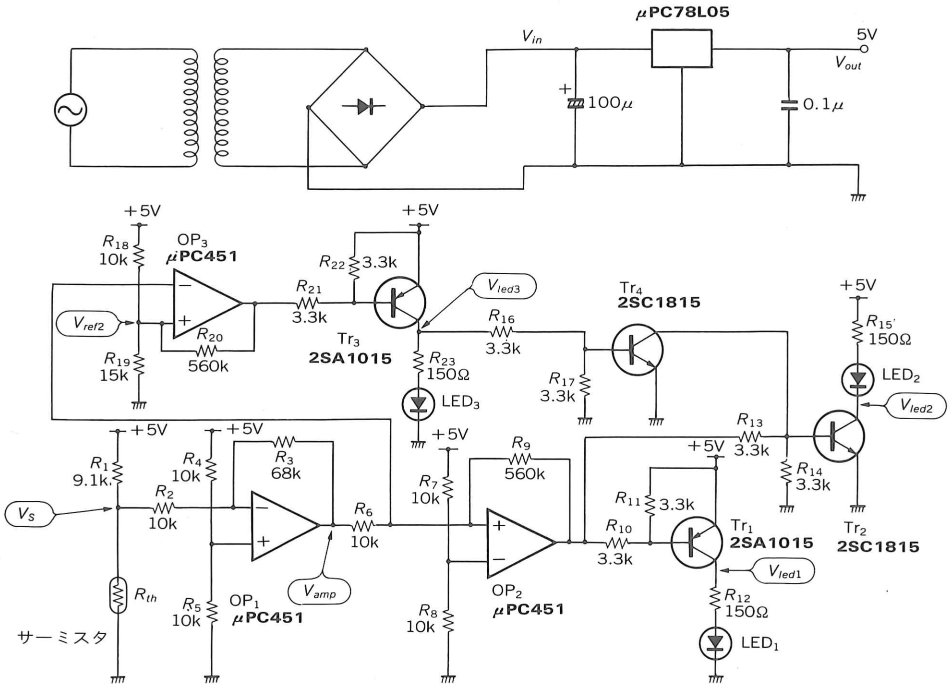


・特殊関数
・条件文(if 文)

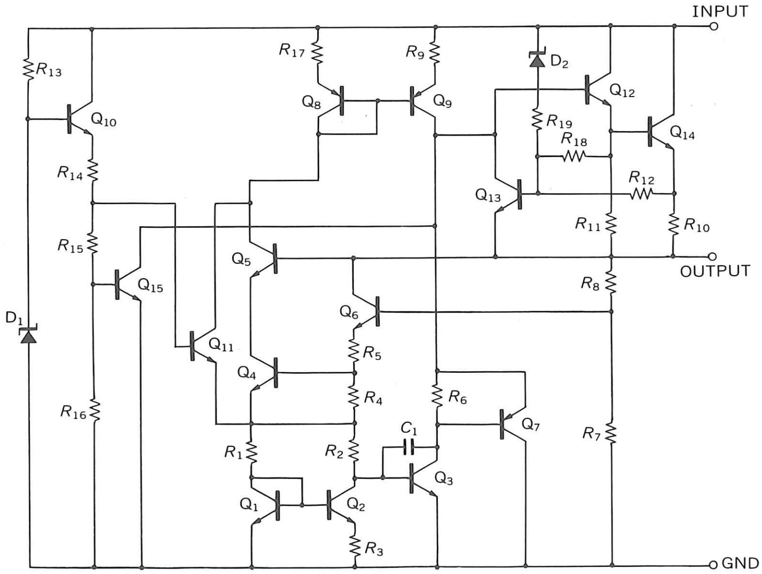
を用いて定義します.

代入式，数式では，加減算など図1-3 の数式が，特殊関数は図1-4 の関数が使用でき，さらに条件文は図1-5 のように使用できます．図1-1 で述べたように NAND ゲートの if 文のモデル作成などに使用するとよいでし

〈図1-6〉 室内温度表示回路



〈図1-8〉[3] μPC78L05 の内部回路



ょう．

記述内容によっては，「define block」を定義しなくてもよい場合がありますが，詳しくは以下の実例で説明します．

それでは次に，ビヘービア・モデルの実例を回路シミュレータによる解析結果例もまじえて解説することにします．

● 室温表示回路

図1-6は室内の最適温度を表示する回路です．各家庭にもエアコンが普及していますが，この回路は3個のLEDを使用して室内の最適温度を表示します．

LED_1が点灯しているときは冷えすぎです．また，LED_3が点灯しているときはエアコンがきいていない状態です．LED_2が点灯しているときは，室内が最適な温度であることを示しています(図1-7)．

図1-6は最適温度を27℃から29℃と定めて設計した回路です．電源は商用電源を整流して，3端子レギュレータでDC 5Vの直流電源を作っています．

温度センサには，サーミスタを使用します．ここでは温度センサからの出力を片電源の反転増幅回路で増幅し，コンパレータ(OPアンプをコンパレータとして使用)で27℃(OP_2)と29℃(OP_3)を判別しています．

さて，図1-6の回路をシミュレーションする場合，最初に突き当たる問題が3端子レギュレータの回路モデルです．どうしたら回路シミュレータで解析ができるようになるでしょうか．

前述のように，等価回路(図1-8)をサブサーキットで定義する方法があります．しかし，この方法では回路が複雑になるうえ，回路定数や半導体名もよくわかりません．

そこでビヘービア・モデルで定義することにします．

3端子レギュレータの回路モデル

ビヘービア・モデルで回路モデルを作成する場合，もっとも重要なことはその素子がどんな動作をするか，そしてどの回路特性が重要な特性かを把握することです．

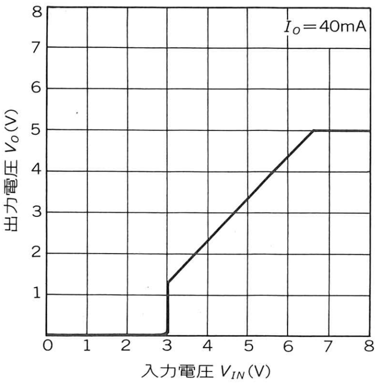
このためには，素子のカタログを熟読し，どこの特性を使おうとしているのかを判断します．

その中にはシミュレーションする際に重要な項目もありますし，さほど重要な項目でないものもあります．この重要度は回路によっても，あるいはどんな回路解析を行うかによっても変わります．

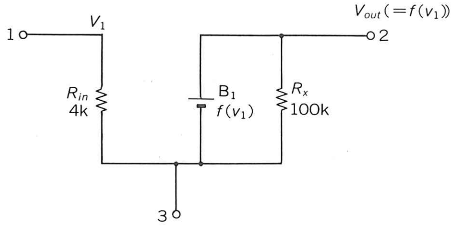
たとえば，図1-6の3端子レギュレータの場合，ざっと次の項目が上げられます．

(1) 入力電圧-出力電圧特性
(2) 温度-出力電圧変化特性
(3) 入力電圧-回路動作電流特性

〈図1-9〉[3] μPC78L05の入出力特性



〈図1-10〉 3端子レギュレータの回路モデル



(4) 周波数-出力インピーダンス特性
(5) 入力変動-出力電圧変化特性
(入力変動過渡応答特性)
(6) 出力電流-出力電圧変化特性
(負荷変動過渡応答特性)

これらの特性も入力電圧や温度，負荷，周波数などいろいろな要素が絡んで変化しますし，また上記にあげなかった特性もあります．

しかし，商用電源を使用するので周波数は50 Hzか60 Hzに限定され，また大きな入力変動や負荷変動も考えられないため(4)～(6)はあまり関係しないと考えられます．図1-6の回路を考えると(1)～(3)あたりがもっとも重要でしょう．

すなわち，この場合の回路モデルは入力電圧と周囲温度に対する出力電圧，そして入力インピーダンスや出力インピーダンスが回路を解析するうえでの重要な要素と考えられます．

そこで，図1-9に入力電圧-出力電圧特性を示します(それ以外の特性の掲載は省略します)が，入力電圧が3Vまでは出力電圧は0Vです．そして入力電圧が6.5V以上で，出力電圧が5Vになっています．

また，マニュアルによると回路動作電流は入力電圧が10Vのとき2.5mAなので，これを抵抗で表すとすれば，その値はおよそ4kΩと考えられます．そして，出力インピーダンスは周波数が数kHzまでは0.1Ωとなっています．

以上の条件から作成した回路モデルが図1-10ですが，消費電流の値から算出して R_{in} を 4 kΩ，出力インピーダンスを 0 Ω としています．

また，B_1 はノード 1－3 間の電圧 V_1 で制御される電圧制御型電圧源で，

$V_1 \leqq 3$ のとき，$f(v_1)=0$

$3 < V_1 \leqq 6.5$ のとき，$f(v_1)=V_1-1.5$

$V_1 > 6.5$ のとき，$f(v_1)=5$

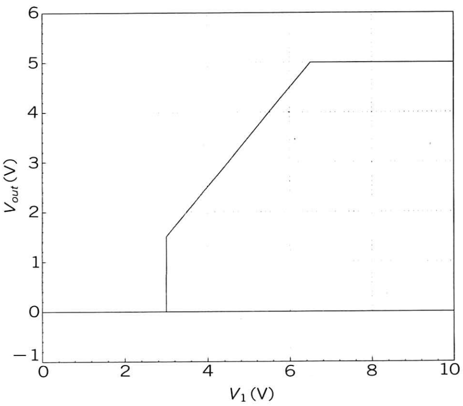
とすれば，図1-9 の特性を満足できます．

図1-10 の回路モデルにしたがい，サブサーキットで定義したファイルがリスト1-1 です．t3_reg がサブサーキット名で，1～3 の数字は図1-10 のノード番号と対応しています．

リスト中に記述する B＊＊＊は電圧制御型電圧源を意味し，ここでは B1 としました．この B1 のラインの意味は，ノード 2－3 間の電圧 Vout はノード 1－3 間の 1 個の電圧源で制御されることを表しています．

そして，「define block」の中に前述の関数が書かれています．その他は SPICE のサブサーキットを作成する場合と同じです．

〈図1-11〉〈リスト1-1〉のシミュレーション結果



〈リスト1-1〉3 端子レギュレータの回路モデル

```
.SUBCKT t3_reg 1 2 3
*Subject:
Rin 1 3 4K
B1 2 3 (v) Vout (1) 1 3
define
Vout(V1)
     if(V1>3.0){
          if(V1>6.5)
               Vout=5.0;
          else
               Vout=V1-5.0;
     }
     else
          Vout=0.0;
exit
Rx 2 3 100K
.ENDS t3_reg
```

このサブサーキットの入力 V1(ノード 1 3 間の電圧)を 0 V から 10 V までスイープさせ，DC 解析で出力電圧(ノード 2 3 間)を調べた結果が図1-11 です．カタログどおり(図1-9)の入力電圧-出力電圧特性が得られ，目的の 3 端子レギュレータの回路モデルができたことになります．

● 3 端子レギュレータの温度特性

ところで，リスト1-1 のサブサーキットには温度特性が加味されていません．

当然のことですが，常温の時と低温時/高温時とはわずかですが特性に変化があります．そこで，今度はこの温度特性を含んだ回路モデル(サブサーキット・ファイル)を作成してみましょう．

カタログによると 25 ℃ のとき，出力電圧の typ 値が 5 V で，出力電圧温度変化は 0.4 mV/℃ となっています．したがって，たとえば周囲温度が 25 ℃ のとき，出力が 5.0 V なら 50 ℃ のときは 5.01 V となります．また，－25 ℃ のときなら 4.99 V です．

すなわち，0.4 mV＊(周囲温度 －25)の温度変化があることになります．このシミュレータでは周囲温度を表す変数として TEMPDC(摂氏温度)が用意されています．

TEMPDC を使って温度特性を追加したサブサーキット・ファイルがリスト1-2 です．式を追加するとともに B1 の電圧制御型電圧源の定義文にパラメータとして TEMPDC を書き加えます．

また，「define block」内の出力電圧 Vout は，入力電圧 V1(ノード 1 3 間電圧)と TEMPDC の関数であることを追記します．

そして温度特性を追加したサブサーキットの V1 を 10 V に固定して温度をパラメータにし，DC 解析を行った

〈リスト1-2〉温度特性を加えた 3 端子レギュレータの回路モデル

```
.SUBCKT t3_reg 1 2 3
*Subject:
Rin 1 3 4K
B1 2 3 (v) Vout (1) 1 3 TEMPDC
define                     周囲温度を表す変数
Vout(V1)
     if(V1>3.0){
          if(V1>6.5)
               Vout=5.0+0.0004*(TEMPDC-25);
          else                 温度ドリフト
               Vout=V1-5.0;
     }
     else
          Vout=0.0;
exit
Rx 2 3 100K
.ENDS t3_reg
```

結果が図1-12 です.

また，TEMPDC と同様に時間や周波数などを記述するための特殊変数もあります(図1-13).

3 端子レギュレータのビヘービア・モデルによる回路モデルがほぼできあがったところで, 図1-6 の電源回路部分のみをシミュレーションしてみます.

TR 解析(時間をスイープさせる)による結果が図1-14 です.

● ワースト・ケースでの解析

これで 3 端子レギュレータも解析できるようになりましたが, リスト1-2 の回路モデルは typ 値を採用して作成しました. しかし，量産する場合などにはワースト値がどうなるか調べておきたいものです.

たとえば，カタロクでは 25℃ のときの出力電圧は,

min：4.8 V
typ：5.0 V
max：5.2 V

とあります.

このため，リスト1-2 の回路モデルも min 値，typ 値, max 値と 3 種類の回路モデルあるいはパラメータを整備しておくと便利です.

言うまでもありませんが，min 値の回路モデルでは出力電圧(リスト1-2 の 5.0 の部分)を 4.8 とし, 温度変化値も出力が最小になる値を設定します. また，max 値の回路モデルでは出力電圧を 5.2 とし，温度変化値も出力が最大になる値を設定します.

図1-6 の回路では周波数を考慮する必要はまったくないので，回路モデルには付加しませんでした.

しかし，回路によっては周波数特性が必要な場合もあると思います. このような場合には変数 FREQ を使用して周波数特性を追加します.

また, リスト1-2 のモデルは負荷がいくら大きくなっても 5 V が供給できるモデルになっています. しかし，実際には出力電流が大きくなると電源電圧は低下します. 図1-6 の回路は負荷電流が小さいのでこのモデルで問題ありませんが，問題となる回路もあります.

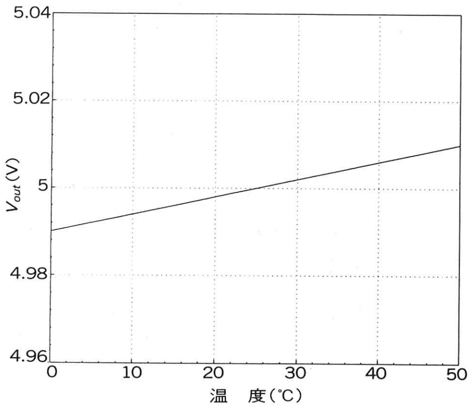
● モデリングする範囲

結局のところ，回路モデルが正確であればあるほど回路設計時に気を使う(シミュレーションでなく頭で判断する)ことが少なくなります. しかし, 回路モデルの作成に時間がかかるうえ，シミュレーション時間も長くなります.

ところが，ラフな回路モデルだとモデルの作成時間もシミュレーション時間も短くなりますが，逆に頭で判断するところが増えます.

どちらが良いかは回路仕様や用途など，さらには個人差により一概に言えません. 多少無責任な意見になりますが，自分でその時の条件で判断するしかないということです.

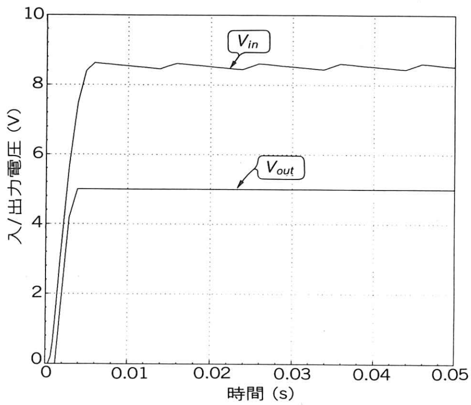
〈図1-12〉 3 端子レギュレータの温度ドリフト(V_1=10 V)



〈図1-13〉 ビヘービア・モデルで使用できる主な特殊関数

TEMPDC	周囲温度(℃)
VT	熱電位 kT/q(k はボルツマン定数，T は絶対温度，q は電子の電荷の絶対値)
TIME	時間(sec)
FREQ	周波数(Hz)

〈図1-14〉 ＴＲ 解析の結果

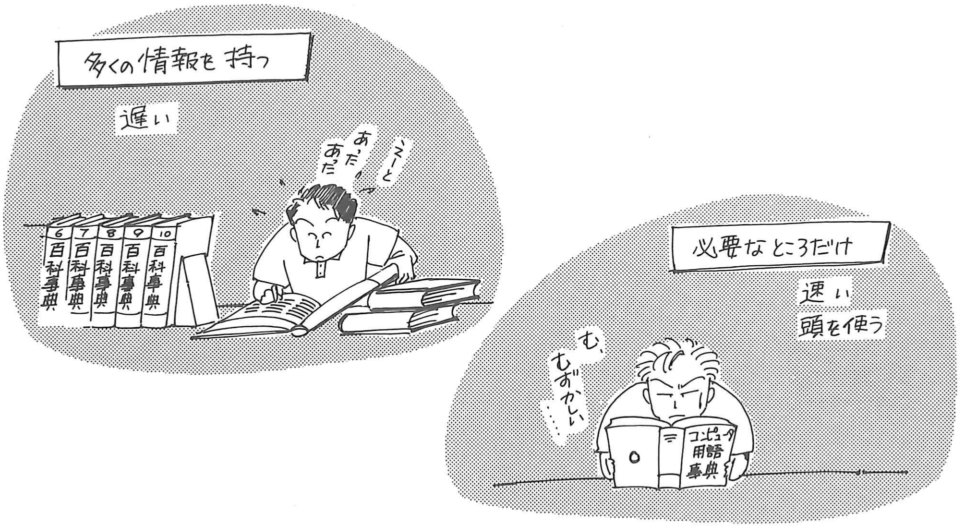


筆者の場合は，多少頭で判断することが増えても回路モデルの作成時間を短くし，そしてシミュレーション効率を上げる方法を取るようにしています.

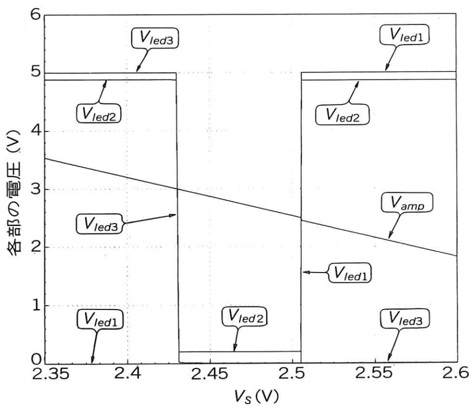
なぜなら，以前の実験や手計算に頼っていたころを考えると, 現状の回路 CAE の機能や能力が大きな労力の削減をもたらしてくれるため，真の回路設計のための考える時間を多く取れるようになったからです.

また，回路モデルがあまりにも複雑になると回路モデルの信頼性が問題になります. 特に，回路設計の際, 思った特性がでないときなどは回路モデルを疑いたくなってしまいます. そうなると，本来なら回路を設計するのが主であるのに，いつの間にか回路モデルを作

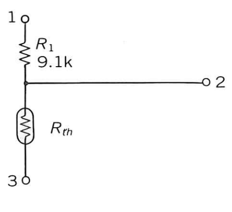
〈図1-15〉 目的に合わせてモデルを作成する



〈図1-16〉 V_Sをスイープさせたときのシミュレーション結果



〈図1-17〉 サーミスタのモデル



〈リスト1-3〉 サーミスタのサブサーキット

```
.SUBCKT thermistor 1 2 3
*Subject:
R1 1 2 9.1K
Rth 2 3 10K*EXPN{4000*{1/{273.15+TEMPDC}-1/{273.15+25}}}
.ENDS thermistor
```

成するのが主になってしまうからです(図1-15).

だからと言って,現状の回路CAEに満足しているわけではありません.もっと性能向上を望みたい点も数多くあります.とりわけ,回路モデルやモデル・パラメータの精度や量は重要です.

サーミスタのモデル

電源部分が解析できるようになったので,次は制御回路部のシミュレーションです.

この場合もサーミスタ(温度センサ)の回路モデルがないので,簡単にはシミュレーションができません.そこで,R_{th}すなわちサーミスタのところに抵抗を入れ,抵抗値をスイープしてDC解析を行う方法がまず思いうかびります.

また,V_sのところにダミー電源を接続し,電圧値をスイープしてシミュレーションを行う方法もあります(図1-16).図からもわかるように確かに基本的には動作しているようです.

しかし,27℃以下でLED_1,27℃から29℃でLED_2,29℃以上でLED_3が点灯しているかどうかは判断できません(当然ですが).

と言うのも,温度センサの出力に代わる仮の入力を設定しているからです.したがって回路動作を確認するためには,このデータから周囲温度が何度になっているかを手計算で算出しなければなりません.

こんな方法では設計に時間がかかります.また,精度にも不安があります.

そこで,図1-17のようにサーミスタのモデルを作成することにします.図のように回路は単純ですが,サ

ーミスタをどう定義するかがポイントになります．サーミスタの温度と抵抗の関係は，一般に次式で表されます．

$$R=R_{25}\exp B(1/T-1/T_{25})$$

ただし，R は $T(K)$ の時の抵抗値

周囲温度 T については前述した変数TEMPDCが使えるので $T=273.15+$TEMPDC と置き換えられます．

また，B 定数は4000のものを使うとすると，図1-17はリスト1-3のサブサーキットになります．リストからわかるように，ここでは「define block」は設定していません．

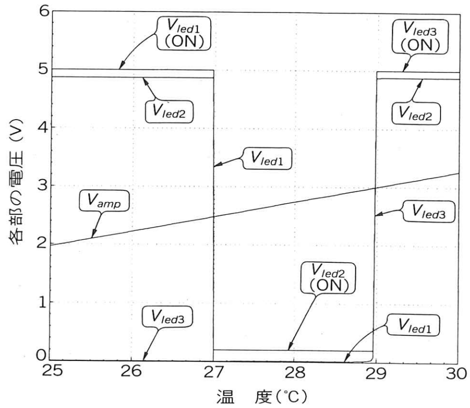
R_{th} は単なる数式(TEMPDCの関数)として表現しているので，シミュレータは通常のネット・リストだと判断します．このために「define block」の設定が必要なくなるわけです．

ところで，サーミスタには自己発熱があり，このための温度上昇が検出誤差を生じます．そこで，これらの目安のための定数として，熱放散定数があります．

ここでは7 mW/℃のものを使用しているので，7 mWの電力を消費するとサーミスタは1℃上昇します．ここでは，R_1(9.1 kΩ)の抵抗がシリーズに接続されているので，消費電力は1 mW以下に抑えられています．

このため，回路モデルには熱放散定数のファクタは組み込んでありません．したがって，サーミスタ単体として(直列抵抗がないとき)モデルを作成する場合には熱放散定数のファクタを組み込んでおいたほうが無難です．

〈図1-18〉最終的な回路動作



さて，回路モデルがなかった3端子レギュレータと温度センサのモデルが完成したところで，全体の回路をシミュレーションしてみます．センサが温度を検出するので，ここでは温度をスイープさせDC解析を実行します．

結果が図1-18ですが，27℃以下で LED_1，27℃から29℃で LED_2，29℃以上で LED_3 が点灯していることがわかります．

◈ 参考・引用*文献 ◈

(1)*テキサスインスツルメンツ，The Bipolar Digital Integrated Circuits Data Book 1985.

(2) 苗手英彦；CAEで学ぶOPアンプ回路入門，CQ出版㈱.

(3)*日本電気㈱，産業用リニアIC データ・ブック 1990.

後記

最近，将棋の世界では，棋士が最高の実力を発揮できる年齢，いわゆる全盛期の年齢が下がっているそうです．パソコンを駆使して膨大な棋譜の管理や指し手の検索を行い，効率よく研究を進めたほうが有利ということになれば，やはりファミコン世代に分があるらしく，その結果として年齢が下がっているのだそうです．

また，ソフトが充分強くなったチェスなどではパソコン・ソフトとの対戦で実力を養い，大会初出場でいきなりトップ・クラスに入ってしまうことなどもあるようです．このような世界でも(だからこそ？)コンピュータが必需品になっているわけです．

そのようなわけで，コンピュータと切っても切れない縁で結ばれている電子技術の世界で，生産までの全工程をコンピュータで行うことができるまでになったのは当然の結果(？)でしょう．

過去の資産はデータ・ベースで楽々管理，実験基板の組み立てもなく，シミュレーションで回路の動作がわかるとなると，これまで以上にアイデアのための時間が取れるようになります．これからは独創と工夫を競い合う世界がやってくるのでしょうか．大いに期待できるところです．



例解 VHDL プログラミング①

構造と機能の表現

長谷川 裕恭

VHDL 言語は高い記述能力をもつ言語である反面，プログラミング言語に不慣れな回路設計者にとっては理解するのに難しい言語だと言われています．VHDL の文法書はいくつか販売されていますが，どれも文法の説明に重点をおき，回路設計者の VHDL 導入には不適切だと思われます．なぜかというと回路設計者にとっては VHDL を覚えることが仕事ではなく，VHDL を使用して効率のよい回路設計をすることが仕事であるはずだからです．

トランジスタ技術 1993 年 3 月号の「ハードウェア記述言語 VHDL 入門」では簡単な文法紹介を行いましたが，その続編として本稿ではシンプルな VHDL 記述をいくつか紹介し，その記述から実際に出力される回路とを対比させ，VHDL を少しずつ理解できるように構成してみました．習うよりも慣れよという言葉のとおり VHDL にまず慣れてください．なお，リストを読む手助けとして，それぞれのリストには，同時にゲートでの表現を掲載します．

〈表 1 - 1〉VHDL の記述レベル

構造化記述	階層設計の記述，ゲート・レベルの記述
論理式記述	“and”や“or”など，論理式で構成された記述
ビヘービア記述	回路の動作，ふるまいを表す記述

VHDL の基本的な記述

● 論理式記述

回路合成のための VHDL 記述は一般的に表 1 - 1 の三つのレベルに分けられます．

リスト 1 - 1 の記述はハーフ・アダーの記述で論理式レベルの記述にあたります．

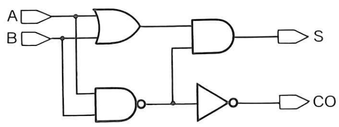
VHDL は①の入出力など外部とのインターフェースを記述したエンティティと，そのエンティティの機能を記述するアーキテクチャ③に分かれます．②はデータ・タイプと呼ばれるもので，本稿では“std_logic”とそのベクタ・タイプ“std_logic_vector”を使用していきます．④は信号宣言で，回路内部だけで使用する信号“C”，“D”を定義しています．⑤が論理式の記述で，論理式には“and”，“or”，“nand”，“nor”，“xor”，“not”の五つがあります．そして信号に値を代入するには，“<=”を使用します．

〈リスト 1 - 1〉ハーフ・アダーの記述(論理式記述)

```
library IEEE;
use IEEE.std_logic_1164.all;

entity HALF_ADDER is
                                        ┐
    port (                              │
        A,B  : in std_logic;            │ ①：エンティティ宣言
        S,CO : out std_logic);          │   入出力ポートを定義する
                   ↑                    │
end HALF_ADDER;                         ┘
              ②：データ・タイプ

architecture DATAFLOW of HALF_ADDER is            ┐
                                                  │
signal C,D : std_logic;←                          │
                   ④：信号宣言                    │
begin                  内部のネットの宣言         │
                                                  │
    C  <= A or B;   ┐                             │ ③：アーキテクチャ宣言
    D  <= A nand B; │ ⑤：論理式の記述             │
    CO <= not D;    │                             │
    S  <= C and D;  ┘                             │
          ↖                                       │
            ⑥：信号代入文                         │
end DATAFLOW;  値を代入する                       ┘
```



〈リスト 1-2〉ハーフ・アダーを使用したフル・アダー(構造化記述)

```
library IEEE;
use IEEE.std_logic_1164.all;

entity FULL_ADDER is
    port (
        A,B,CIN : in  std_logic;
        S,CO    : out std_logic);
end FULL_ADDER;


architecture STRUCTURE of FULL_ADDER is

component HALF_ADDER
    port ( A,B  : in std_logic;     } ①:コンポーネント宣言
           S,CO : out std_logic);   }   使用する下位モジュールを定義
end component;

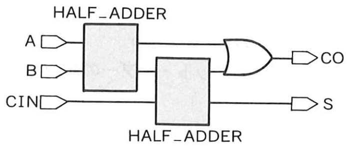
signal U0_CO,U0_S,U1_CO : std_logic;

begin

    U0 : HALF_ADDER port map ( A,B,U0_S,U0_CO );   } ②:インスタンス文
    U1 : HALF_ADDER port map ( U0_S,CIN,S,U1_CO);  }   下位モジュールを呼び出す

    CO <= U0_CO or U1_CO;

end STRUCTURE;
```



● 構造化記述

リスト 1-1で記述したハーフ・アダーを二つ使用してフル・アダーを記述したのが**リスト 1-2**になります．この記述の①と②の部分が構造化記述にあたります．

構造化記述とは，階層設計で下位のモジュールを呼び出したり，あるいはロジック・ゲートをじかに使用する記述を言います．

VHDL では②のようにインスタンス文で下位のモジュールを呼び出す前に，①のように“architecture”と“begin”の間にそのモジュールがどのような形で構成されているかを表すコンポーネント宣言をしなければなりません．

▶ ベクタの記述

データ・バスなど幅を持った信号は**リスト 1-3**の①のようにデータ・タイプ“std_logic”のベクタ・タイプである“std_logic_vector”にバス幅を定義します(②)．ここで使用している降順の“downto”の反対に昇順の“to”を使用することもできますが演算上不都合をおこすことがあるので普通は“downto”を使用すると考えてください．

③の論理演算子“and”と“or”は幅を持った信号の場合，1ビットごとの演算になります．この場合，論理演算子“＋”の右辺と左辺，そして代入される値はすべて同じビット幅でないとエラーになります．

● ビヘービア記述(算術演算の記述)

ビヘービア記述について明確な定義はありません．

通常，論理式記述よりも抽象度が高いということで，比較演算子(<,>,=,<=,>=)や算術演算子(+,−,*,/,mod, rem)を使用した記述，process 文(後述)を使用した記述を言います．

リスト 1-4の④の記述は算術演算子“＋”があるだけの簡単な記述ですが，結果として大量のロジックを生成することに留意してください．

なお，現状の回路生成ソフトでは，(/,mod, rem)は右辺が2の累乗の定数(実質的にはビット・シフト演算になる)の場合しか回路を生成しません．

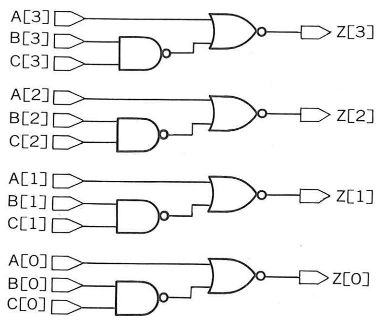
ビット型のベクタ・タイプである“std_logic_vector”

〈リスト 1-3〉データ・バスの記述(ベクタの記述)

```
library IEEE;
use IEEE.std_logic_1164.all;

                                                ②:ビット幅を定義
entity SAMP1 is
    port ( A,B,C : in  std_logic_vector(3 downto 0);
           Z     : out std_logic_vector(3 downto 0));
end SAMP1;
                                      ①:データ・タイプ std-logic-vector
                                         std-logicのベクタ・タイプ
architecture DATAFLOW of SAMP1 is
begin

    Z <= A nor (B nand C); <---------- ③:ベクタどうしの論理式(nor, nand)
                                          の右辺と左辺および代入される信
end DATAFLOW;                             号は同じビット幅でなければなら
                                          ない
```



〈リスト1-4〉
4ビット・アダーの記述(ビヘービァ記述)

```
library IEEE; ←────────①：ライブラリの指定

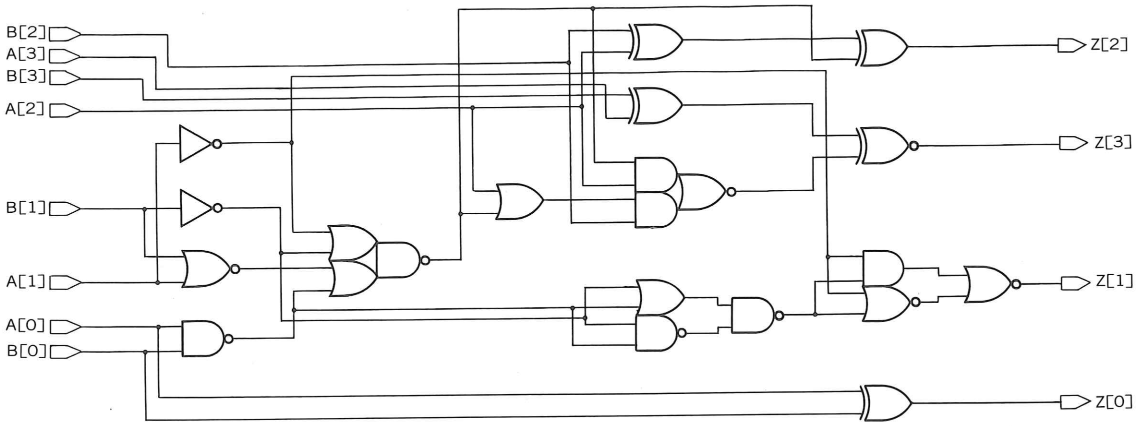
use IEEE.std_logic_1164.all; ←──②：IEEE-1164で定められたパッケージ
                                   std-logicを使用するためにはこの
                                   パッケージのロードが必要

use IEEE.std_logic_arith.all;    } ③：std-logic-vectorに算術演算
use IEEE.std_logic_unsigned.all; }     をさせるためのパッケージ


entity ADDER4 is
    port ( A,B   : in  std_logic_vector(3 downto 0);
           Z     : out std_logic_vector(3 downto 0));
end ADDER4;


architecture DATAFLOW of ADDER4 is
begin

    Z <= A + B; ←──④：加算の記述
                      代入される信号は
                      同じビット幅でなければならない
end DATAFLOW;
```



〈リスト1-5〉
4ビット・コンパレータの記述
(if文)

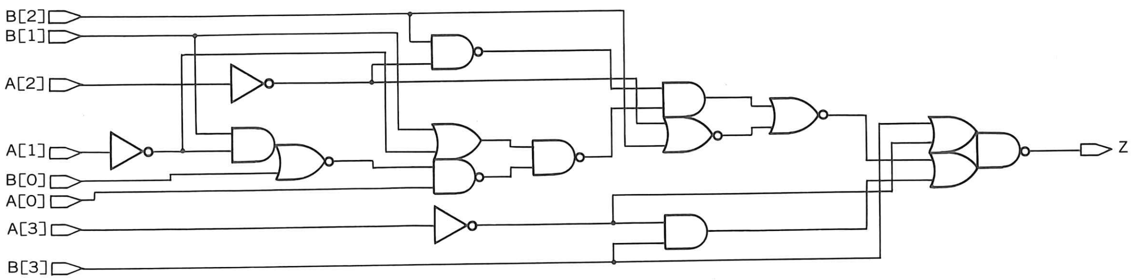
```
library IEEE;
use IEEE.std_logic_1164.all;

entity COMP4 is
    port ( A,B : in  std_logic_vector(3 downto 0);
           Z   : out std_logic);
end COMP4;


architecture RTL of COMP4 is
begin

    process(A,B) begin                              ┐
              ↖──②：センシティビティ・リスト        │
       if(A > B) then                               │
             ↖──③：式                               │
                                                    │ ①プロセス文
          Z <= '1'; ←── ④：値は' 'でくくる         │
       else                                         │
          Z <= '0';                                 │
       end if;                                      │
                                                    │
    end process;                                    ┘


end RTL;
```



は，そのままでは算術演算をすることができません．算術演算をさせるためには“std_logic_vector”の算術演算を定義したパッケージ(データ宣言や function の集まり)をロードする必要があります．この場合は③がそれにあたります．

ただし，ここで呼び出しているパッケージ“IEEE. std_logic_arith”と“IEEE. std_logic_unsigned”は，シノプシス社から提供されているものです．いままで，“std_logic”と“std_logic_vector”という二つのデータ・タイプを使用してきましたが，このデータ・タイプはパッケージ“IEEE. std_logic_1164”で定義されているもので，このパッケージをロードしないと使用することができません．

このパッケージは IEEE-1164 として今年 VHDL の仕様に追加されるものです．パッケージをロードする場合は，その前に①のようにそのパッケージが納められているライブラリを記述しておく必要があります．パッケージの後の“.all”はそのパッケージの中身をすべて呼ぶという意味になります．

VHDL の制御文(プロセス文)

● if 文の記述

if 文，case文，for 文を使用したより高度な記述をする場合には，プロセス文を使用します．**リスト 1-5** の①がそれにあたります．

プロセス文の始め，②の文に“( )”で囲われた部分がありますが，これはセンシティビティ・リストと呼ばれ，この部分に書かれた信号のいずれかが変化したときにプロセス文の“begin”以下を実行するしくみになっています．

組み合わせロジックを生成させるためには，このセンシティビティ・リストにすべての入力信号を記述する必要があります．

“std_logic”などのビット型のデータ・タイプに値を代入する場合は，④のように値を' 'でくくる必要があります．

if 文は③の式が真(TRUE)のときに then 以下を実行し，偽(FALSE)のときに else 以下を実行します．

比較演算子は他の演算子と異なり，論理値のデータ・タイプ boolean の値(TRUE または FALSE)を返すので，“std_logic”や“std_logic_vector”などで宣言された信号に直接代入することはできません．通常，比較演算子は，if 文と組み合わせて使用することになります．

● case 文の記述

リスト 1-6 は case 文を使用して 3 ビット to 8 ビットのマルチプレクサを記述したものです．

①の“&”はビットを連接させる演算子で，1 ビットの信号を 3 ビットに連接して“INDATA”に代入しています．

ベクタ・タイプ“std_logic_vector”の値は②のように" "でくくります．

case 文はすべての値について記述しないと文法エラ

〈リスト 1-6〉 3 ビット to 8 ビット・マルチプレクサの記述(case 文)

```
library IEEE;
use IEEE.std_logic_1164.all;

entity MUX3TO8 is
    port ( A,B,C : in  std_logic;
           Y     : out std_logic_vector(7 downto 0));
end MUX3TO8;

architecture RTL of MUX3TO8 is

signal INDATA : std_logic_vector(2 downto 0);

begin

   INDATA <= A & B & C; ←———— ①：ビットの結合

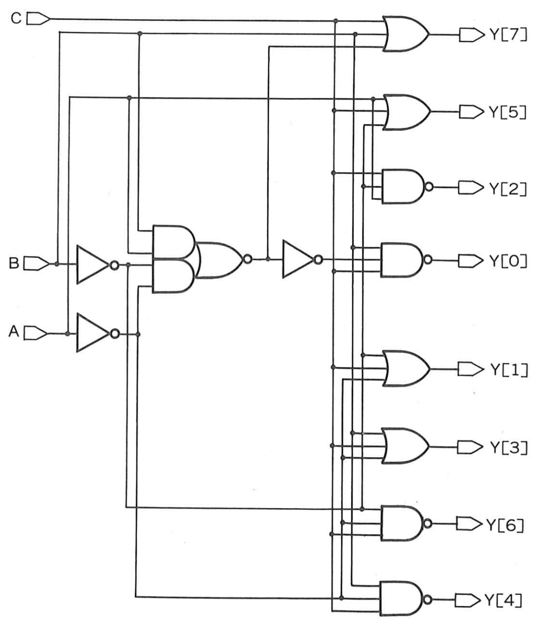
   process(INDATA) begin

      case INDATA is

         when "000" => Y <= "01111111";   ┐
         when "001" => Y <= "10111111";   │
         when "010" => Y <= "11011111";   │
         when "011" => Y <= "11101111";   │
         when "100" => Y <= "11110111";   ├ ② case文
         when "101" => Y <= "11111011";   │
         when "110" => Y <= "11111101";   │
         when "111" => Y <= "11111110";   │
         when others => Y <= "XXXXXXXX";  ┘
                 ↖
      end case;     ③：残りのすべて

   end process;

end RTL;
```



〈リスト1-7〉 5ビット・パリティ・チェッカの記述(for 文)

```
library IEEE;
use IEEE.std_logic_1164.all;

entity PARITY_CHECK is
    port ( A      : in  std_logic_vector(4 downto 0);
           Y      : out std_logic);
end PARITY_CHECK;


architecture RTL of PARITY_CHECK is

begin

   process(A)

   variable TMP : std_logic;◀—①：変数宣言

   begin

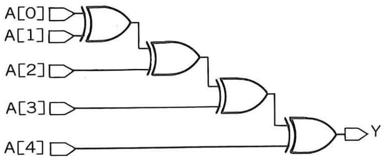
      TMP := '0';

      for I in 0 to 4 loop            } ②：for文の記述
          TMP := TMP xor A(I);
      end loop;                 ③：Iビット目の信号の取り出し

      Y <= TMP; ◀— ④

   end process;

end RTL;
```



ーになります．データ・タイプ “std_logic” は，‘1’, ‘0’ という値のほかに不定 ‘X’, ハイインピーダンス ‘Z’ などの値があり，これらは回路生成上は意味がありませんが VHDL の文法上，記述する必要があります．

③の “others” とは残りすべての値という意味で，‘1’, ‘0’ 以外の値の記述をこの1行によって省略しています．

● for 文の記述

リスト1-7の①は，変数宣言になります．

信号宣言がアーキテクチャ宣言内でするのに対し，変数宣言はプロセス文の内部で行い，プロセス文内のローカルな変数になります．変数宣言された変数は，“:=” によって数値が代入されます．

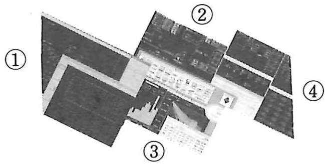
for 文は規則的な繰り返しがある場合に用います．②の記述は “xor” の繰り返しに for 文を使用しています．ベクタから1ビット取り出す場合には③のように “( )” の中に何ビット目を取り出すか記述します．変数宣言された “TMP” はプロセス文の中でのみ使用されるローカルな変数なので④のように信号 “Y” に代入し直して出力します．

◈参考文献◈

(1)IEEE Standard 1076, VHDL Language Reference Manual, IEEE, 1987.
(2)Steve Carlson；VHDL による HDL 設計と論理合成入門, Synopsys, 1990.
(3)VHDL 言語ワークショップ, Synopsys, 1993.
(4)R. Lipsett, Carl Schaefer, ary Ussery；VHDL：Hardware Description and Design, Kluwer Academic Publishers, 1989.
(5)長谷川裕恭：ハードウェア記述言語(HDL)を使ったロジック設計法，トランジスタ技術，1992年11月号，p313.
(6)長谷川裕恭：ハードウェア記述言語 VHDL 入門，トランジスタ技術，1993年3月号，p288.

表紙について

今回の表紙は以下の製品の画面写真によって構成しました．御協力いただきました各メーカに対しこの場を借りて厚く御礼申し上げます．



● 画面の構成

① ACLAB
横河ディジタルコンピュータ㈱
☎0422(56)9359

② OrCAD
インテリジェント・システムズ・ジャパン㈱
☎045(633)2634

③ CR-3000/AWS
㈱図研
☎045(942)1511

④ Artist DANTES
ケイデンス・デザイン・システムズ㈱
☎045(475)2211



仕様書などはしっかり読んでおこう！

コンパイラの仕様にない事項

辻　秀之

● なぜ高級言語を使うか

すべてのソフトウェアをアセンブラで開発しようとすると，おそらくたいへんなコストがかかるでしょう．このため，必要な部分だけにアセンブラを使い，それ以外は高級言語で開発するというのが一般的です．

高級言語で開発することによって少々ソフトウェアの性能が落ちたとしても，それによって節約できたコストの一部をハードウェアに振り向ければ，たいていの場合はお釣りがくるはずです．

本稿では，高級言語がソフトウェア・エンジニアだけではなく，ハードウェア・エンジニアにも使われるようになる今日このごろの傾向にあたり，高級言語とのつき合い方について少し解説を加えます．

高級言語の仕様

どんな高級言語であろうと文法が規定してあり，その文法で規定されている範囲でならばコンパイラが違っても処理される内容は同じです．文法で規定されていない事柄については何の保証もありません．したがってそのようなソース・プログラムを二つの異なるコンパイラで機械語に変換させた場合，両者の動作がまったく同じになるという保証はありません．

〈仕様に書かれていないことは…〉



これは通常では問題になりませんし，仮に問題になるとしてもプログラマがそれを承知していて，仕様書にここはこれこれの理由で問題になるかもしれないと明記してあるのならかまわないと思います．

しかし，プログラムを作った人がそれに気がついていない場合には深刻なバグが発生したり，バグでないものをバグだと思い込んで悪戦苦闘したりするかもしれません．

つまりは仕様をしっかりと理解しているか，いないかという問題に還元されるわけです．

もちろん，高級言語の文法を規定するときにすべての動作を厳密に決めてあればこのようなことは起こらないはずですが，それでは困ることもあります．

たとえば2バイトのデータをメモリに格納するとき，上位のバイトと下位のバイトのどちらを先に置くかはCPUによって違います．このため，ほとんどの高級言語では2バイトのデータの順番については特に規定せず個別のコンパイラの作成者にまかせています．

当然，コンパイラの設計者はターゲットCPUに都合のよい順序でデータを扱うようにコンパイラを設計します．このためメモリを直接操作するプログラムを書けば両者の動作は異なることになりますし，事実そのようなプログラムは高級言語で書かれていても機種に依存してしまうので互換性はありません(そういうプログラムを作るのがいけないと言っているわけではありません．ただし，ソース・プログラムに注意書きをしておくくらいの配慮は必要だと思います)．

さて，互換性を維持するため，高級言語の規格を作るときにこの2バイトの順序をどちらかに統一したらどうなるでしょうか．

2バイト・データの順序はCPUによって違うのですから，統一した順序とは逆になってしまうCPUが必ず存在してしまいます．

この不運なCPUでは2バイト・データに関する機械語命令はすべて使えなくなってしまい，1バイト単位で順序を並べ替えながらメモリを読まなければなりま

せん．そうなればソフトウェアの性能は著しく低下するでしょう．

結局，世の中にある多くのCPUに対して個別に都合のよいコンパイラを作れるようにするには，ある程度あいまいな部分を持たせておいたほうがよいということです．

あいまいであるというよりは，わざと規定せずコンパイラの設計者に選択の余地を残してあるのだと考えたほうがよいかもしれません．

ただし，高級言語のどの部分をわざと規定しないかは，それもまた規定されています．規格の原典には書かれているはずです．

式の評価順序

数式の出てこないプログラムはないと思いますが，たとえば，

$$x=\sin(a)+\sin(b)$$

のような式を計算させる場合，常識的には先に書いてある $\sin(a)$ の計算をするだろうと思えます．しかし，どちらを先に計算しても結果として得られる x の値は同じです．

数式がたくさん出てくるプログラムでは，もしかしたらこの式の手前にある別の式にも $\sin(b)$ があり，すでに計算済みかもしれません．

この場合，よくできたコンパイラなら，以前に計算しておいた $\sin(b)$ の値を保存しておき，この式で再利用するような機械語プログラムを生成するかもしれません(もちろん b の値が変わっていないことをチェックして)．

また，複数のコプロセッサをもっていて，三角関数を同時にいくつも計算できるコンピュータもないわけではなく，その場合には $\sin(a)$ と $\sin(b)$ を同時に計算させるのがよいかもしれません．

このようなことが許されるためには数式の評価順序に関する規定はあいまいでなければなりません．もし，必ず前から順に計算しなければならないのだと決めてしまったら，上記のような工夫は禁止しなければなりません．

実際，最もユーザが多いと考えられるC言語を例にとると，演算子の優先順位(＊ は ＋ より優先されるなど)は明確に決まっていますが式の評価順序は自由です．

そして，数式の評価順序が決まっていないということは同じ数式でもコンパイラによっては計算順序が違うかもしれないということです．

その簡単な例を**リスト1**に示します．

リスト1にある int f1(int a)という関数は引き数 a を画面に表示し，そのまま a の値を返してくるだけの関数です．引き数の値を表示してくれるので，どの順番で f1(1)～f1(3) が実行されているかは画面を見ればわかります．

もう一つの関数 float f2(int a) は f1 とほとんど同じですが，整数型で受け取った引き数を浮動小数点に変換して返します．

これを Boland C＋＋ バージョン 3.00 と，Microsoft C バージョン 6.00A，そして HI-TECH C バージョン 7.11(Z80 用)でコンパイルして実行させた結果を図1に示します．コンパイラによって計算の順序が違っていることがわかります．

三つのコンパイラで結果がすべて違っている，x＝f1(1)＋f1(2)＋f1(3) について見てみましょう．

Boland C＋＋ の結果は常識的な順序ですので特に触れません．

Microsoft C では，3，2，1 という逆の順序，HI-TECH C ではもっと奇妙な 2，1，3 という順序で関数を評価(実行)しています．

もちろん，Microsoft C や HI-TECH C にバグがあってこうなったのではなく，これも正しいのです．

さらに詳しいことはアセンブル・リストを見るとわかります(**リスト2**，**リスト3**)．アセンブル・リストを順に追っていくと Microsoft C では関数 f1(a) を呼び出す順序は逆でも，足し算「＋」の順序は普通の順序(左から右)であることが読み取れます．

つまり，3，2，1 の順序で関数を計算して，その結果は 1＋2，1＋2＋3 の順で計算しているのです．

一方，HI-TECH C では 2，1，3 の順に関数を計算し，足し算は 3＋2，3＋2＋1 のように逆の順序(右から左)です．

なぜ，関数や足し算の計算順序を変えているのかはわかりませんが，考えられる利点の一つとしてはスタックを上手に使うことができるということがあります．

スタックは先に入れたものが後に取り出されるので，後ろから入れていかないと取り出すときに前から取り出せない(前の方から計算できない)わけです．

またコンパイラが数式を解析するときにもスタックのような処理を使っているので，そこでも順序が入れ替わる可能性があります．

おもしろいことに浮動小数点を使った式，y＝f2(1)＋f2(2)＋f2(3) ではまた違った順序で関数を呼び出しています．

● 順序が変わって困ること

f1(1) が先に計算されようが f1(3) が先に計算されようが結果は同じ(この場合 6)になるのですから，ほとんどの場合は問題ありません．

では，どういう場合に問題になるかというと，この場合もそうですが関数が単に数値を返すだけでなく，何か別の動作(この場合では画面に数字を表示する)を

〈リスト1〉サンプル・プログラム

```
#include <stdio.h>

int f1(int a)
{
    printf("%d",a);
    return a;
}

float   f2(int a)
{
    printf("%d",a);
    return (float)a;
}

void main()
{
    int x;
    float y;

    x = f1(1)+f1(2)+f1(3);
    printf("%d",x);

    y = f2(1)+f2(2)+f2(3);
    printf("%f",y);
}
```

〈図1〉実行結果

```
1 ⎫ f1(1)+f1(2)+f1(3)
2 ⎬ の計算順序
3 ⎭
6
1 ⎫ f2(1)+f2(2)+f2(3)
2 ⎬ の計算順序
3 ⎭
6.000000
```

(a) Boland C++ 3.00

```
3 ⎫ f1(1)+f2(2)+f1(3)
2 ⎬ の計算順序
1 ⎭ 逆になっている
6
1 ⎫ f2(1)+f2(2)+f2(3)
2 ⎬ の計算順序
3 ⎭
6.000000
```

(b) Microsoft C 6.00A

```
2 ⎫ f1(1)+f1(2)+f1(3)の計算順序
1 ⎬ f1(2)が最初に計算されている!
3 ⎭
6
3 ⎫ f2(1)+f2(2)+f2(3)
2 ⎬ の計算順序
1 ⎭
6.000000
```

(c) HI-TECH C 7.11(Z80用)

する場合です.

例えば，f1(a)がa番目の機械を操作し，その結果を返すという関数だったとします.

1番目の機械と2番目の機械を動かし，その結果の合計値が欲しいときは，

x1=f1(1); x2=f2(2); x = x1+x2;

と書けば，1番目の機械が必ず最初に動き，次に2番目の機械が動きますが，もしこの3行を一つにまとめて，

x=f1(1)+f1(2);

と書いてしまったらどちらの機械が先に動くかはコンパイラにまかせることになります.

それで困るかどうかはプログラムを作る人が判断すべきことで，どちらが正しいとか間違いだとかの問題ではありません.

ただ，こういうこともあるのだということを，知っていないと何かのときにI/Oポートを読む順序がおかしいのはハードのバグだろうかなどと悩むことになるかもしれません.

同様な話としてインクリメント演算子「++」の付いている変数が一つの式に2回以上使われている場合のあいまいさがあります.

たとえば，x [0]，x [1] ...をy [0]，y [1] ...にしてやろうと考えて，プログラムの中でx[i]=y[i++]とかx[i++]=y[i]などと書くとわけのわからないことが起きるかもしれません(最近のコンパイラにはこういうプログラムを書くと式があいまいだと警告してくれるものもあります).

これは，どこがあいまいなのでしょうか．x[i]=y[i++]について考えてみます.

メモリ上にあるx [i]という変数にアクセスするにはアドレスを計算しなければなりません.

x [i]のアドレスは(x [0]のアドレス)+i(この変数の語長)となります.

右辺y [i++]ほうは，まずy [i]のアドレスが計算されてからi++が計算されます.この順序は保証されています.

しかし，x [i]のアドレス計算はi++より先に実行されるかもしれないし，後かもしれません．この順序は決まっていません.

したがって，x [i]=y [i++]という代入文はコンパイラしだいで結果が変わってしまうのです.

しかも恐ろしいことにi++のあとでx [i]のアドレスが計算される場合にはiの値が1つ増えていますので場合によってはx [i]の範囲を越えて次のアドレスにある別の変数に代入してしまうこともあります.

別の変数が書き換えられるくらいで済めばまだましで，C言語の場合はスタック上に変数領域を取りますから最悪の場合にはスタックの内容が破壊され暴走する可能性もあるのです.

これは，関数の評価の例と同様に記述を2行に分けてx [i]=y [i];i++と書けば解決します.

また，実用性はないかもしれませんが，最初にk=i

〈リスト2〉Microsoft C 6.00A のアセンブラ出力

```
    fstp    qword ptr [bp-24]    ; スタック経由で printf関数に値を渡す

_main   PROC NEAR
    push    bp
    mov bp,sp
    mov ax,14
    call    __aNchkstk
    push    di
    push    si
;   x = -2
;   y = -6
; Line 22
    mov ax,3
    push    ax              ; スタック経由で関数に値を渡す
    call    _f1             ; f1(3) の計算
                          ; 結果は AXレジスタに戻される
    add sp,2                ; スタックの調整
    mov cx,2
    push    cx
    mov si,ax               ; f1(3) の結果は SIに保存する
    call    _f1             ; f1(2) の計算
    add sp,2
    mov cx,1
    push    cx
    mov di,ax               ; f1(2) の結果は DIに保存する
    call    _f1             ; f1(1) の計算
    add sp,2
    add di,ax               ; f1(1)+f1(2) の計算
    add si,di               ; これで、f1(1)+f2(2)+f1(3)が得られた
    push    si              ; スタック経由で printf関数に値を渡す
    mov ax,OFFSET DGROUP:$SG175
    push    ax              ; 同上。文字列 "%d"
    call    _printf
    add sp,4
; Line 25
    mov ax,1
    push    ax              ; スタック経由で関数に値を渡す
    call    _f2             ; f2(1) の計算
                          ; 計算結果はメモリに入っていて、その
                          ; アドレスがAXレジスタに返される。
    add sp,2                ; スタックの調整
    mov bx,ax
    fld DWORD PTR [bx]      ; f2(1) の結果をコプロセッサのスタック
                          ; に読み込む。
                          ; (またはエミュレーションライブラリが、
                          ;  同様な処理をする)
    mov ax,2
    push    ax
    fstp    DWORD PTR [bp-10]   ; コプロセッサのスタックにある f2(1)の
                          ; 値を一時変数に保存する
    call    _f2             ; f2(2) の計算
    add sp,2
    mov bx,ax
    fld DWORD PTR [bx]      ; f2(2) の結果をコプロセッサのスタック
                          ; に読み込む。
    fadd    DWORD PTR [bp-10]   ; f2(1)+f2(2) の計算
                          ; fadd は、コプロセッサ内のスタックと
                          ; メモリにある値を足し合わせる命令
                          ; (またはエミュレーションライブラリが、
                          ;  同様な処理をする)

    mov ax,3
    push    ax
    fstp    DWORD PTR [bp-14]   ; f2(1)+f2(2) の結果を一時変数に保存
    call    _f2             ; f2(3) の計算
    add sp,2
    mov bx,ax
    fld DWORD PTR [bx]      ; f2(3) の結果をコプロセッサのスタック
                          ; に読み込む。
    fadd    DWORD PTR [bp-14]   ; これで、f2(1)+f2(2)+f2(3) が得られた
    fwait
    sub sp,8
    mov bx,sp
    fstp    QWORD PTR [bx]      ; 計算結果をスタック経由で printf関数
                          ; に渡す
    fwait
    mov ax,OFFSET DGROUP:$SG176
    push    ax              ; 同上。文字列 "%f"
    call    _printf
    add sp,10               ; スタックの調整
; Line 26
    pop si
    pop di
    mov sp,bp
    pop bp
    ret
    nop

_main   ENDP
```

〈リスト3〉HI-TECH C 7.11 のアセンブラ出力(抜粋)

```
;C:CTEST.C: 21: x = f1(1)+f1(2)+f1(3);
    ld  de,2        ; DEレジスタで値を渡す
    call    _f1     ; f1(2) の計算
                  ; 結果はHLレジスタに入ってくる
    ex  de,hl
    push    de      ; f1(2) の結果を保存
    ld  de,1
    call    _f1     ; f1(1) の計算
    ex  de,hl
    push    de      ; f1(1) を保存
    ld  de,3
    call    _f1     ; f1(3) の計算
    pop de          ; f1(1) をスタックから戻す
    add hl,de       ; f1(3)+f1(1) の計算
    pop de          ; f1(2) をスタックから戻す
    add hl,de       ; f1(3)+f1(1)+f1(2) の計算

;C:CTEST.C: 24: y = f2(1)+f2(2)+f2(3);
    ld  de,3
    call    _f2     ; f2(3) の計算
                  ; 結果はDE, HLレジスタに入ってくる
    push    hl      ; f2(3) を保存
    push    de
    ld  de,2
    call    _f2     ; f2(2) の計算
    push    hl      ; f2(2) を保存
    push    de
    ld  de,1
    call    _f2     ; f2(1) の計算

    call    fladd   ; f2(1)+f2(2) の計算。
                  ; fladd は、スタック上の値とDEHLレジ
                  ; スタの値(いずれも float)を加算して、
                  ; 結果をDEHLレジスタに返すサブルーチン。

    call    fladd   ; f2(1)+f2(2)+f2(3) の計算
```

としておいて，x[k++]=y[i++]と書くのも大丈夫です．この場合xとyのどちらのアドレスが先に計算されるかはわかりませんが，xのアドレスがk++の前に計算されることと，yのアドレスがi++の前に計算されることは保証されているからです．

C言語は他の言語にはない強力な演算子があり，簡潔な表現で複雑な処理をさせることができます．しかし，この複雑な処理が実行される順序は，すべて決まっているわけではありません．したがって，一つの式の中で実行順序が入れ替わっては困る場合には，複数の式に分割しておく必要があります．

コンパイラの仕様には，しばしばこのような落とし穴的なものもあります．ユーザから見れば，やっかいな問題を引きおこす落とし穴なのですが，前述のように立派な理由があってのものです．パズル的な思考実験なのでうんざりするかもしれませんが，しっかりと理解したうえで高級言語を使うようにしてください．



君は異次元クリエイター

アイデアで勝負する
画像エンジニア募集！

募集職種

①アナログ回路設計技術者
広帯域ビデオアンプ、PLL、AD/DAコンバータ周辺等
②デジタル回路設計技術者
フレームメモリ、CPU及び周辺回路等、主として画像処理関連
③ソフトウェア技術者
UNIX、アセンブラ等のファームウェア開発
④メンテナンス・システムエンジニア
計測機器、画像処理機器等のメンテナンス、営業技術
⑤品質管理・調整検査
電子機器の品質管理、調整検査等

募集要項

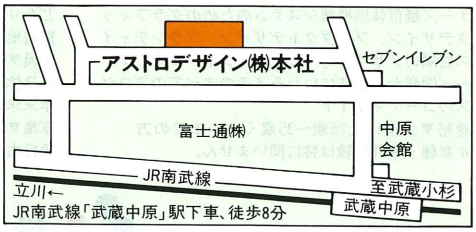
資　　格：30歳位まで
給　　与：18〜35万円、経験者優遇
(例)'92年大卒初任給実績 205,000円
待　　遇：賞与年2回、昇給年1回、交通費全額支給、各種社会保険完備、財形貯蓄、年金基金、退職金共済、所得補償保険、海外研修、社員旅行ほか
勤　　務：9:00〜17:30
勤 務 地：川崎市(希望により、松山・米子も可)
休　　日：完全週休2日制、祝日、年末年始ほか(年間休日121日)
休　　暇：有給休暇、慶弔休暇ほか
交　　通：JR南武線武蔵中原駅下車徒歩8分

応　募

履歴書(写真貼付)および職務経歴書(書式自由)をご郵送ください。
追って面接日をご連絡いたします。(応募秘密厳守)

会社概要

事業内容：映像機器・計測機器の設計・開発・製造・販売
代 表 者：鈴木茂昭
設　　立：昭和52年2月
従 業 員：120名
年　　商：25億円('92年実績)



アストロデザイン株式会社

本　社　神奈川県川崎市中原区上小田中863 〒211 TEL(044)751-1011 FAX(044)751-1300
連絡先　TEL(044)751-9410 総務部採用担当

(年俸制300万〜1,000万円。)

しかし本当のステイタスは自己のパイオニア・スピリットに捧げたい。

当社の社長は、独創的な製品の開発者として知る人ぞ知る存在です。そんな彼が次なるターゲットとして選んだのが、各種ビジネスからアミューズメントまで、その応用の期待されるVIRTUAL REALITY（バーチャル・リアリティー＝仮想現実感）システム。そしてその追求のために設立されたのが当社〝㈱スリーディ〞です。VRという全く新しい分野を手がけようと集まった人間を指して社長は敢えて〝技術屋〞という言葉をつかう。世間一般の卑屈な意味ではなく、そこには技術を愉しめる技術者への彼なりの賛辞が表されているのです。設立〜今日まではVRを市場に定着させるためのいわば準備期間。当社のVRシステム第一弾としてのゴルフシミュレーターの開発を始め、販売形態の確立など着実な足場作りに専念してきました。これらが軌道にのった今、スリーディはいよいよ第2創業期を迎えます。未知なる〝VR世界〞への出航に向けて、その中枢となっていただくスターティングクルーを大募集します。給与は300万〜1000万の年俸制。当社の期待を少しでも感じていただければ幸いです。

(確実な信頼が、ビッグな力を味方につけた。)

しかしそれに甘えはしない。より大きな発展のために。

設立以来の地道な活動の結果を示すのが、当社をとりまく信頼関係。その一つの例がこれからの活動のベースとなる自社開発商品〝ゴルフシミュレーター〞に表れています。あの尾崎将司プロの推奨と、商品PRのための全面的な協力により、この商品がプロの使用に耐える画期的なゴルフシステムであることへのアピールが可能になったのです。二つ目はシステム普及のための販売チャネルに三井物産を迎えたこと。三井の持つ販売のノウハウを背景に、システム単体の販売だけでなく、これらを集めた全く新しいアミューズメントスポット〝ファミリーグリーンランド〞の提案、そしてこれに伴うフランチャイズ（加盟店）システムの展開へと、VR分野におけるトータルな活動が実現しました。もちろん事業だけでなく、社員への待遇面でも画期的な企業になりたいと考えています。先に述べた年俸制の給与体系もその一部。来るべきVR時代のパイオニアを目指すスリーディ。あなたの参加を心からお待ちしています。

VR（バーチャル・リアリティーフロンティア）開拓地へ…。



(3次元FAX PAT取得
上場準備中

募集要項
職種▼①ハードウエアエンジニア／疑似体感機器システムのためのコンピュータグラフィックシステムの開発（機構設計、カスタムICの設計） ②ソフト開発プログラマー／疑似体感機器システムのソフトウエアの開発 ③システムクリエイター／ハード＆ソフトの開発効率を向上させるための各種開発支援システム等の設計・構築 ④デザイナー／疑似体感機器システムのためのグラフィックデザイン、プロダクトデザイン、フランチャイズ店舗のデザイン等 ⑤プロダクト・マネージャー／開発から製造にいたるまでのすべてのプロセスのコーディネイト
資格▼大卒以上25歳〜35歳くらいまでの方 ※業種・業界経験は特に問いません。
給与▼年俸制 300万円〜1000万円 ※年齢や経験年数でなく、能力を正当に評価し、個別に設定します。
待遇▼昇給年1回、賞与年2回、交通費全額支給、各種社会保険完備 ※借り上げ住宅・寮のご相談に応じます。
時間▼9:00〜17:00 ※「残業は翌日の仕事に影響する」と考えている会社ですから、残業はほとんどありません。
勤務地▼東京都大田区南千束1-4-1
交通▼東急池上線長原駅より徒歩2分
休日休暇▼完全週休二日制（土・日・祝）、夏季、年末年始、年次有給休暇
募集▼履歴書（写真貼付）および職務経歴書（書式自由、これまでの職務内容を詳しくお書きください。）を下記宛郵送してください。尚、追って、こちらから連絡をさしあげますので、連絡の比較的とりやすい時間をお書き添えください。 ※入社日は相談に応じます。※面接の日時は便宜をはかります。 ※応募の秘密は厳守します。

会社概要
設立▼昭和59年5月 資本金▼1000万円
代表者▼代表取締役社長 新井容徳
社員数▼10名
売上高▼10億円（平成3年12月実績）
主要取引先▼三井物産、資生堂、鐘紡など
事業内容▼疑似体感機器システムの開発・製造・販売、洗浄システムの開発及び製造（販売は三井物産が担当）

株式会社スリーディ

〒145 東京都大田区南千束1-4-1 コーポートビル3階 TEL.03-3728-3316㈹ 担当／新井

▶資料請求：TM02◀



■募集職種

●電子回路設計技術者

①アナログ回路設計
A/D、D/A、センサー、AV、モーターコントロール、フィルター

②デジタル回路設計
CPU、グラフィック、LAN、通信、メモリー

③高周波回路設計
AV、ビデオ入/出力

④ASIC設計

●ソフトウェア技術者

①ドライバーソフトの開発
C、アセンブラ

②ファームウェアの開発
MS-DOS

③アプリケーションソフトの開発

④信号処理、画像処理プログラムの開発

⑤OS関連の開発

■給与

能力給制度を適用し、経験・能力・前給与などを考慮して優遇いたします。

■年令

30才位まで。

■待遇

昇給年1回、賞与年2回、社会保険完備、資格手当
住宅手当、扶養家族手当、通勤費手当、地域手当。

■勤務時間

9:00～18:00

■休日

週休2日制（平成5年110日、年末年始・GW・盆休み有）
有給休暇・特別休暇

◎応募方法

電話連絡または、履歴書・職務経歴書を送付して下さい。
応募の秘密は厳守致します。また、出社日については相談させていただきます。
本社 総務部 森本まで（082-282-8017直）

Interface
Dream & Freedom
株式会社 インタフェース

本社 〒732 広島市南区南蟹屋1-1-25 TEL.082-282-9801
大阪営業所 〒542 大阪市中央区上汐2-2-11柳原ビル TEL.06-763-4440
東京営業所 〒105 東京都大田区大森北1-22-9コロナ大森ビル5F TEL.03-3766-6181
インタフェース大分 〒873-02 大分県東国東郡安岐町大字下原1428 TEL.09786-7-3300

S・U・T・E・K・I 空間



ヒトはもっとSUTEKIになれる。
夢の空間で、未来をクリエイト！

[ハード設計技術者募集]

【職　　種】①システム設計
　16・32ビットのハード開発
②ASIC設計
　メンターによるASIC設計
　ゲートアレイ設計
　スタンダードセル設計

【資　　格】高卒以上　25～35歳位迄　男女
実務経験者の方

【勤務地】東京都練馬区石神井台〈本社〉

【勤務時間】10:00～19:00

【給　　与】固定給制25～60万円　経験・能力・前職での給与等を考慮の上、当社規定により優遇します。
年俸　①800～1,200万円可能
　　　②500～900万円可能

【待　　遇】昇給年1回〈実力により臨時昇給有〉
賞与年2回〈他、業績により特別賞与有〉
各種社会保険完備　交通費全額支給　団体保険加入　保養所〈房総・草津〉　社員旅行毎年海外／ハワイ・グァム・台湾　海外研修制度毎年実施／アメリカ・ヨーロッパ　利子補給預金制度

【休日休暇】週休2日制／日曜、第1・3・5土曜
祝日年末年始 夏期 有給 特別 慶弔

【設　　立】昭和55年

【資本金】1億2000万円

【年　　商】20億円／平成3年度実績

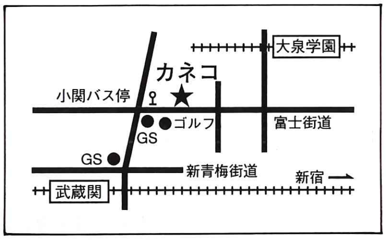
【社員数】70名

【事業内容】コンピューターゲームの企画・開発・製造・販売及び輸出。ハード・ソフト両面からトータルにプロデュースする、業務用・家庭用ゲーム業界屈指の企業です。

【事業所】練馬区石神井台、米国シカゴ、英国ロンドン

【取引先】任天堂、セガ、タイトー、ナムコ、ハドソン、NEC

【応　　募】まずは電話連絡下さい。面接日・入社日等相談に応じます。応募の秘密は厳守します。問合せ歓迎します。



KANEKO 株式会社 金子製作所

本社：東京都練馬区石神井台8丁目23番21号　TEL：03-3921-9661　担当：人事部



●あなたの本音がカタチとなる。

アイデアを実現したい、技術の追究に専念したい。あなたの心の奥底で、エンジニアの本能が暴れたがっていませんか。「俺ならこうする」…。そんなエンジニア魂を存分に発揮できる会社、それが西湘エレクトロニクスです。“普通の企業がやらないこと”をテーマに磁気記録装置を中心とした自社製品開発に取り組み、世界からも高い評価を得ている少数精鋭集団。ソフトウェアと回路設計、さらにメカトロニクスとの技術複合を図り、生産ライン全体の開発・製造を中心に行ない、これから幅広いメカトロニクス企業として飛躍しようとしています。

●個性を活かして、技を磨く。

海あり、山あり、夏は涼しく、冬は暖かい。レジャー施設やスポーツ施設も豊富に揃った快適な街、小田原に西湘エレクトロニクスはあります。エンジニアが各自の自主管理で自由に仕事を進められるように、フレックスタイム制、半日休暇制度を導入。自分の能力・技術力を試し、のばしていける社内環境が揃っています。遠隔地の方にはアパート（6・6・4.5畳、バス・トイレ付き）を便宜。敷金・礼金・引越し費用は会社負担です。小田原の恵まれた環境の中であなたの個性を発揮して、世界に通用する新製品を共に開発しませんか。

▶募集要項◀　職種●ソフトウェア技術者、回路設計技術者、メカ設計技術者、システム設計技術者、セールスエンジニア　資格●年齢35歳位迄　給与●能力・経験を考慮の上、当社規定により優遇。待遇●昇給年1回、賞与年2回、各種社会保険完備、交通費全額支給、住宅手当、皆勤手当、技術手当、扶養手当、決算賞与、社内旅行有　※遠隔地の方には住宅の便宜はかります。(6・6・4.5・バス・トイレ付相当、敷金・礼金・引越し費用会社負担）勤務地●本社（小田原）　勤務時間●9:00〜18:00　フレックスタイム制　休日休暇●週休2日制、祝日、年末年始、有給休暇、特別休暇　応募●電話連絡の上、履歴書（写真貼付）を持参、又は郵送して下さい。※面接日、入社日相談に応じます。※応募の秘密は厳守します。

フレックスタイム制

▶会社概要◀　設立●昭和61年1月　資本金●1億2800万円　事業内容●磁気記録製品および光磁気記録製品の試験装置の開発・販売　ＦＡ関連自動化システムの受託開発

▶採用関係係連絡先◀　本社●〒250 神奈川県小田原市南鴨宮3-45-4
☎0465-47-3771(代)　担当／瀬戸

お気軽にお電話にてお問い合わせ下さい。
資料請求される方は、巻末の資料請求ハガキ（飛竜企画宛）をご利用下さい。

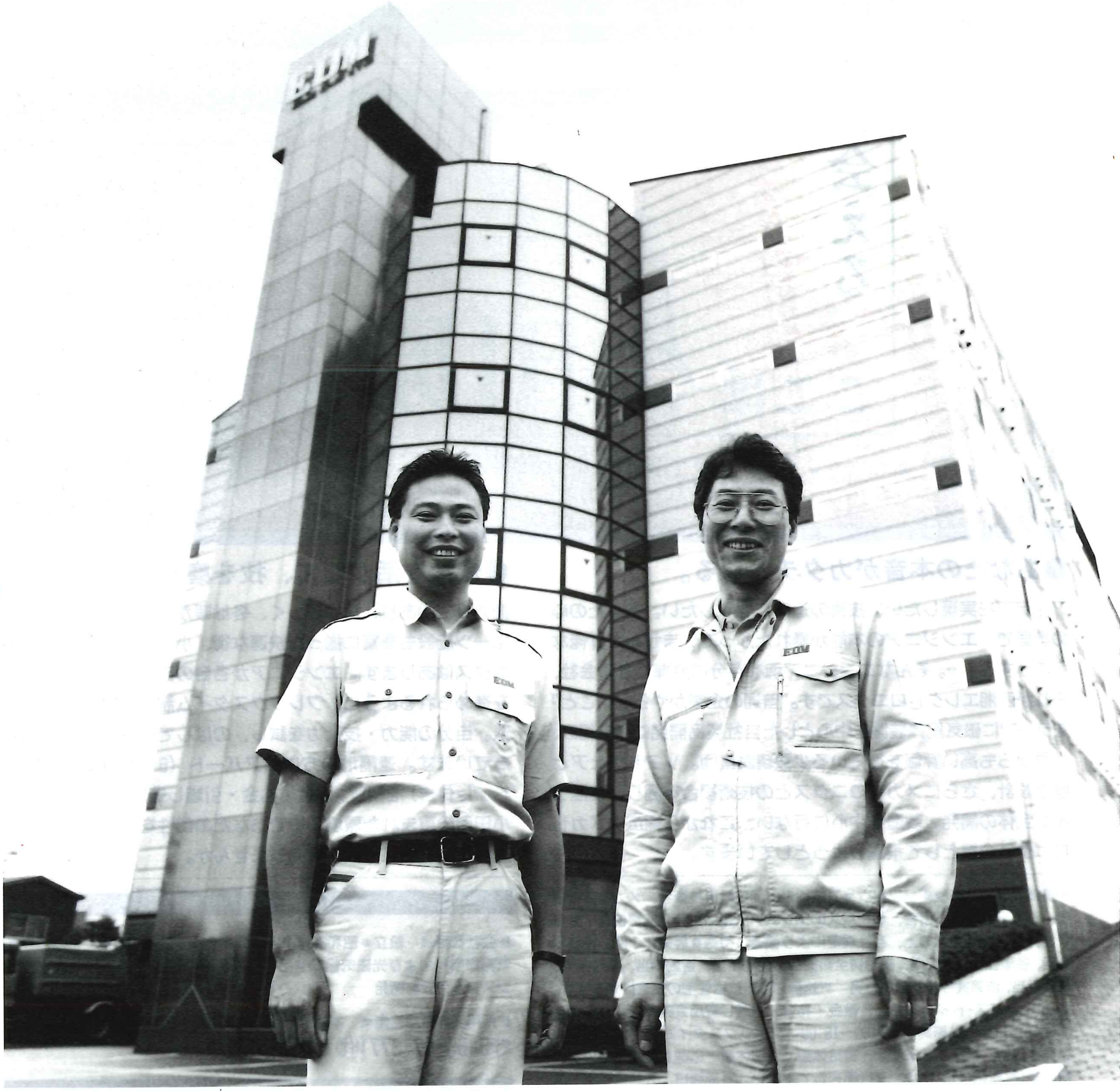
西湘 SEISHO　㈱西湘エレクトロニクス

本社／神奈川県小田原市南鴨宮3-45-4　〒250 TEL (0465)47-3771(代)　担当／瀬戸

▶資料請求：TM03◀

時には、ぶつかることもあ

ＥＤＭの製品は電気設計と機械設計ふたつのセクションの技術を融合させて創られています。
どちらか一方が欠けても、遠慮しても、新しい技術を誕生させることはできません。
言いたいことを言い、納得いくまで話し合う。これがＥＤＭのエンジニアスキルなのです。



神崎 壽人（電気設計）
昭和35年１月１日生まれ
とにかく電気“設計”にたずさわりたくてＥＤＭに入った。アマチュア無線、４級船舶など趣味の分野にも､その行動力を活かしている。昭和60年10月入社。

小野 耕治（機械設計）
昭和30年４月20日生まれ
転職動機はＥＤＭの業績の良さと、技術系企業らしからぬ休日の多さだと言い切る。趣味はテニス･スキーでかなりの腕前らしいとの噂。昭和58年７月入社。

る。

EDM CORPORATION

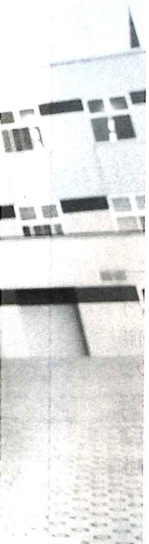
商品の徹底した品質管理が求められる今日、製造年月日・賞味期限などの“表示”は、食品・化粧品・各種工業製品などの生産に不可欠な要素となっています。ＥＤＭはこの“表示”を行なう各種印字機器を始め、印字状態をチェックする“検査機器”や、インクなどの消耗品の開発・設計まで“表示”を核に、ラインの省力化、無人化へのトータルな技術を創り出している企業です。 その開発の中心に位置するのは、電気設計、機械設計のエンジニア。今回は各セクションを代表する２人のエンジニアに登場してもらい、 ＥＤＭの事業内容、 仕事内容、 待遇等について日頃思っていることを語ってもらいます。

電気・機械のコンビで、プランニングから開発を担当。

「プランニングから参加し、工程をトータルに扱えることが嬉しいですね。」と電気設計を担当する神崎さんは語る。「今コンビを組んでいる小野さんとは、もうすぐ10年近いつきあいになります。今の開発事情はどうしても電気設計が先行することが多いので、機械設計の相棒には無理を聞いてもらうことが多いんですよ。」と機械設計の小野さんに目をやる。「確かにセンサー関係や電気ボックスの処理に悩まされることもありますよ。電気設計が先行するってこともあるけど、電気のほうは常に３人程度の機械設計と組んでますからね。それでどうしても応援しちゃうわけ。」と小野さんは笑う。「とにかく、お互い刺激しあいながら仕事を進めています。技術面でぶつかりあいもありますよ。プランニングで半日過ぎることもよくある。でも、決して意地の張り合いじゃない。その証拠に開発中に浮かんだアイデアは随時交換しあいます。セクションが隣同志なので一日に何度も往復するんです。」

エンジニアが創った、エンジニア思いの開発環境。

「ＥＤＭの社長は、日本初の熱転写プリンターを開発したエンジニアでもあるんです。だから、我々の気持ちを本当に良く理解してくれています。開発環境などが良い例だと思いますね。」と神崎さん。「個々がパーテーションで仕切られた開発スペースを持っているんですよ。電気の方はＣＡＤ化されています。機械の方は相変わらずドラフターを使ってるけどね。」と神崎さんの方をにらむ小野さん、同僚というよりまるで悪友のようだ。「教育の仕方だっていわゆるトップダウンとは違う。例えばうちのエンジニアはハード畑出身者ばかりだけど、みんなソフト開発も出来るんです。やれって言われるからやるんではなく、やっぱり自分の設計した電気回路を、自分の手でテストランしてみたいって気持ちが先に立つんですよ。」と神崎さん。アナログ、デジタル、ＣＰＵソフト、シーケンサーと幅広い技術を持つＥＤＭのエンジニア。ぶつかりあいながら成長する彼らに、向かうところ敵はなさそうだ。



■会社概要■

設立●昭和45年8月14日　資本金●3000万円　売上高●50億円（平成5年4月実績）　従業員数●220名　事業所●埼玉テクノセンター　営業所●東京、名古屋、大阪、福岡　海外オフィス●ソウル　主要取引先●山崎製パン、伊藤ハム、サントリー、大正製薬、資生堂、キヤノン、本田技研工業、日本たばこ産業、その他6000余社　事業内容●表示機器（ＦＡ用プリンター・レーザーマーキングシステム・ラベリングマシン）、印字検査機等の開発、設計、販売、輸出入。及び副資材（カーボンテープ・インク・タックシール印刷）、委託加工等の製造・販売

■募集要項■

職種●制御回路設計エンジニア
資格●24歳～35歳位迄　リレーシーケンス・シーケンサー・デジタル回路のいずれかの設計開発経験者
給与●25万円～38万円　※経験・年齢・前給を考慮の上、優遇いたします。
休日休暇●完全週休２日制、祝日、夏季、年末年始、有給休暇、慶弔休暇　※年間休日110日（平成5年度）
勤務地●埼玉テクノセンター（埼玉県比企郡川島町大字戸守392-11）
勤務時間●9:00～17:00
待遇●昇給年1回、賞与年2回（平成4年実績5.5ヵ月）、通勤交通費全額支給、各種社会保険完備、財形貯蓄、退職金制度（退職年金加入）、社宅有、時間外・家族・役職手当を別途支給、社員旅行（国内・海外）、会員制レジャー施設加入、クラブ活動（野球・テニス・ゴルフ・スキー他）
応募●電話連絡の上、履歴書（写真貼付）と職務経歴書（書式は自由ですがなるべく詳しくお書きください）を面接時にご持参ください。郵送も受け付けています。その場合は追って面接日等詳細を通知いたしますので、希望連絡先を履歴書欄外にお書きください。※応募の秘密は厳守します。※面接日・入社日相談に応じます。※人材登録制度があります。今すぐ応募できない方はご相談ください。
本社までの交通●地下鉄都営三田線板橋区役所前駅下車徒歩3分　JR埼京線板橋駅下車徒歩10分　東武東上線下板橋駅下車徒歩10分
勤務地までの交通●東武東上線若葉駅下車送迎バスにて15分（車通勤可）

巻末の共通資料請求ハガキ（飛竜企画行き）をご利用ください。

イーデーエム株式会社　〒173 東京都板橋区板橋3-5-2
TEL 03-3964-4006 担当／塩原・神長

▶資料請求：TM04◀



みんな、ひとりひとりが個性の違う人間。
だから、自分のオリジナルな個性の部分を
もっと大切にしてほしい。
型破りのアイデアとは、えてして、そういうトコロから
生まれることが多いんだ。
ソフトウェアからハードウェアまで
システム開発には、入念な情報収集とともに
優れたアイデアへの発想力が必要。
常識による固定概念にとらわれるばかりではなく、
フリースタイルの発想を描ける
柔軟な思考回路を持つ君たちを
三幸電子は期待しています。

会社概要

■設立/昭和34年7月■資本金/9,600万円■事業内容/マイクロコンピュータシステム開発・設計・製造●職種/電子回路設計エンジニア/プログラマー/生産技術/ソフトウエアエンジニア/営業●資格/高卒以上30歳位迄、経験者優遇●給与/経験・能力により優遇●待遇/昇給年1回、賞与年2回、交通費全額支給各種社会保険、退職金、役職手当●休日・休暇/日曜祝日、GW、夏季、年末年始(年間125日)●勤務地/本社(天白区)オペレーションセンター(日進町)●応募/総務山口へ電話連絡の上、履歴書(写真貼付)持参又は郵送

SANKO ELECTRONICS 三幸電子

〒468 名古屋市天白区植田3丁目802番地 TEL(052)805-2111(代)

▶資料請求：TM05◀

▶資料請求：TM06◀

ALINCO

技術者、募集。



アルインコは「社会に奉仕」、「会社の発展」、「社員の成長」を社憲にアマチュア無線機や業務用無線機器のメーカーとして成長し続けている会社です。当社はこのたびアルインコ研究開発センター開設および業務拡張のため無線通信機器のソフトウェアー開発技術者や設計開発技術者を求めております。皆さまの積極的なご応募をお待ちいたしております。

●求人内容

1. ソフトウェアー開発技術者
（無線通信機、各種電子機器の制御プログラム開発）
2. 設計開発技術者
（アマチュア無線通信機、一般用無線通信機器、業務用各種無線通信機器、デジタル方式通信機器の設計開発）
3. 無線機器調整・検査技術者
（各種無線通信機器の調整・検査）

●応募資格

学歴／当社は実力主義ですから経験者については学歴・学部門等は問いませんのでどんどんご応募ください。来年の春に大学または専門学校を卒業する方も歓迎します。

年齢／一応40才以下としますが、40才以上の方でも経験豊かな方はご応募ください。

●待　遇

経験年数や現在の年収、専門資格などを考慮のうえ待遇します。

●勤務場所

大阪／大阪市中央区城見ツイン21・MIDタワー23F
アルインコ㈱　大阪本部
東京／東京都豊島区池袋サンシャイン60・22F
アルインコ㈱　東京本部
富山／富山県射水郡小杉町手崎493
アルインコ富山㈱

●勤務時間

大阪、東京／AM9：00〜17：30 完全週休2日制

●応募方法

履歴書（写真貼付）に希望職種を明記したもの一通と職務経験や専門資格の有る方は別紙にその内容を詳しく書いて、大阪本部または東京本部人事課宛にご郵送ください。追って面接日・面接場所などをご連絡いたします。今すぐ入社できない方もご応募ください。入社日は相談に応じます。なお、応募の秘密は厳守しますのでご安心ください。また、応募書類は返却いたしませんのであらかじめご了承ください。

■会社概要

創　立／昭和13年9月
資本金／3億5千407万円
年　商／321億円
（平成4年3月期アルインコグループ）
従業員数／700名（アルインコグループ）

■会社の業務

1. 無線通信機および周辺機器の製造販売
2. 建設用機材の製造販売およびリース
3. DIY用品、スポーツ用品の製造販売

アルインコ株式会社

本社・大阪本部／〒540 大阪市中央区城見2丁目1番61号ツイン21・MIDタワー23階

HUMO



八王子で技を磨く、選別機にとことんこだわる。

一台ごとに新製品、終わりのない技術革新。

テレビ、無線電話機など多種多様の電化製品に組み込まれている"水晶振動子"。ヒューモはその選別システムの開発を中心に、コンデンサ・セラミック部品など様々な電子部品の自動選別機を生み出しています。エレクトロニクス分野の進歩に伴い、電子部品は高品質・小型化へと精度を上げていくので、ヒューモの技術力も常に新しさが要求されます。斬新な企画力、そして創造力の発揮・新開発へ挑戦を続ける技術者のための会社、それがヒューモです。

技術者の喜びを追求できる開発環境。

西八王子工場の従業員は12人。うち三人のチーフクラスの技術者が、一人あたり三台ほど開発を受け持っています。発注を受け、図面を起こし、製品をテストして納品するまでおよそ三ヵ月。ひとつの製品開発を広い視野で見渡せ、自分の技術力をまんべんなく発揮できます。もちろん取引先へメンテナンスに赴くことや、チーフとしての管理能力が問われることもあり、単純に技術だけに没頭できるわけではありません。ヒューモでは、リーダーとしての責任感を持ちながら仕事にこだわれる人材を望んでいます。入社して半年間、ヒューモの技術とチームワークをしっかり理解できれば、技を磨ける喜び、キャリアアップの喜びが待っています。

会社概要

設立●昭和48年2月
代表者●玉貫勝美
平均年齢●32歳
従業員数●61名（男46名　女15名）
事業所●本社（東京都杉並区）
　工場（西八王子・浜松）
資本金●5500万円
売上高●9億8000万円（平成4年度）
主要取引先●東芝・日本電波工業・京セラ・松下電子部品・大真空・キンセキ・東洋通信機・三菱マテリアル他
事業内容●計測制御専用マイクロコンピュータ及び各種半導体・電子部品の全自動検査装置の開発・設計及び販売

募集要項

職種●制御用回路設計管理者
資格●30歳～35歳位
　※経験10年以上歓迎します
給与●経験・年齢・能力・前給等を考慮の上、当社規定により優遇します。
　35歳例）月給33万7000円
　※給与面でも納得していただけるように、徐々に改善していこうと思っています。
休日●完全週休2日制、祝日
休暇●夏季、年末年始休暇有(年間休日118日)、ゴールデンウィーク
待遇●昇給年1回、賞与年2回、通勤交通費支給、職務手当あり、社会保険完備、退職金制度有り
勤務地●西八王子工場
勤務時間●8:31～17:20
福利厚生●伊豆・山中湖に健保組合の保養所有、リゾートクラブ法人会員、昼食無償支給
応募●履歴書（写真貼付）を本社宛ご郵送下さい。面接の日時を後日連絡いたします。
交通●ＪＲ中央線西八王子駅下車徒歩５分

株式会社
ヒューモラボラトリー

〒167 東京都杉並区西荻北5-19-11
☎(03)3395-5311㈹　担当●高橋・岩淵

▶資料請求：TM07◀

SACOM SYSTEM SACOM
RYOGOKU SAKURAI-BLDG 38-16 4 CHOME RYOGOKU SUMIDA KU TOKYO 130 JAPAN
TEL 03 635 5145 FAX 03 635 5148



待っているだけじゃダメだ。

『……そうかもしれない』

将来的な不安を感じながら、望みが叶うのをじっと待つ。
ただひたすらに我慢して、チャンスが来るのをじっと待つ、
待つ…。果たしてこれで、本人の理想とする
ステイタスアップが図れるのでしょうか。
もしあなたが、不本意ながらこんな状況に
追い込まれているのなら、
はやくシステムサコムの存在を見い出してください。
自分の発想が活かせ、可能性を自由に伸ばしていける当社なら、
あなたの〝これから〟を大切に育てられるはずですから…。

「貢献宣言」 あなたと共に、世界に貢献する システムサコム　　募集広告 第98弾

●募集職種	●職務内容	●資格
①デジタル回路設計 ファームウェア ソフトウェア	AI(人工知能関連機器開発) 各種ボードコンピュータ 画像処理システム 音声認識システム 計測システム パソコン周辺機器 ネットワーク関連ユニット FDデュプリケートシステム	性別／男性・女性 年齢／20歳～35歳 経験／2年以上 学歴／高卒以上
②生産技術	試作・調整・量産設計・生産管理	

■会社概要■
設立／1981年12月
資本金／1000万円
売上高／12億円（平成4年5月）
社員数／50名

■業務内容■
①ハードウェア関連
コンピュータ周辺装置・FA関連装置・産業設備装置・AI関連システム・ファームウェア
②ソフトウェア関連
・マルチメディアソフト企画制作
・エンターテイメント企画制作
・電子出版企画制作
③ビジネスソフト関連
オフコン・汎用機業務用ソフト・UNIX業務用パッケージ・パソコン業務用パッケージ(NetWare・MS-Window対応)

■主要取引先■
富士通㈱、日本電気㈱、ソニー㈱、富士通OA㈱、ソフトバンク㈱、㈱ソフトウィング、松下電器産業㈱、㈱アスキー

■募集要項■
給与●25歳例／27万円
待遇●昇1、賞2、交全給、各社保完、住宅・残業・皆勤・精勤手当有
勤務地●
本社（墨田区両国）
横浜開発センター（平沼橋）
勤務時間●
9:00～17:30(両国)
10:00～18:30(横浜)
休日休暇●完全週休2日制(土・日)、祝日、夏季、有給、慶弔
応募●電話連絡の上、履歴書(写真貼付)持参　※郵送も可

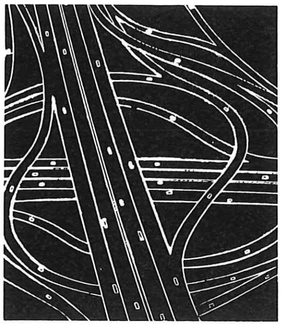
株式会社 **システムサコム**
〒130 東京都墨田区両国4-38-16 両国桜井ビル4F

0120-44-9918
お気軽にお電話下さい。
TEL.03(3635)5145　担当／人事部 伊佐、当山

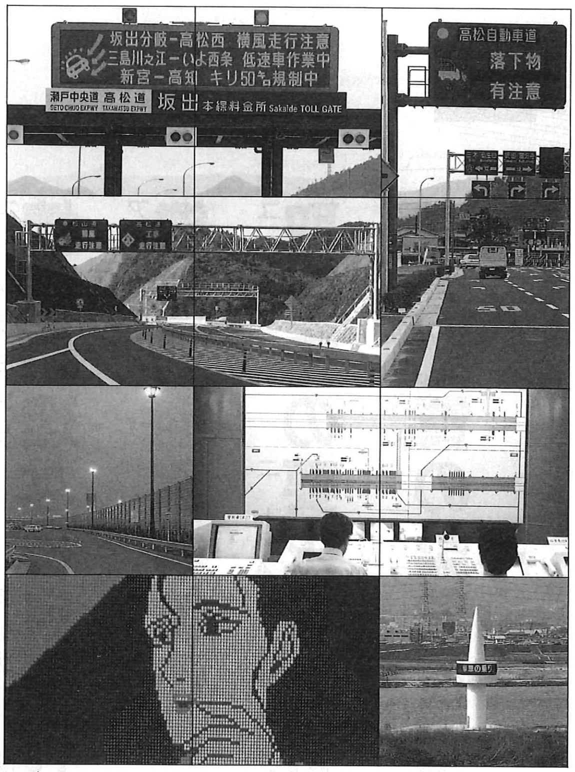
▶資料請求：TM08◀

光と情報のテクノロジー

安全を技術でおとどけする星和電機は道路情報表示装置及びLED表示方式の多目的カラーディスプレイの開発メーカーです。また京都における次期リーディング産業として期待の高いソフトウェア技術、メカトロニクス技術及びそれらに関連する科学技術分野の振興と発展に寄与する為に創られる「京都高度技術研究所」に参加し、長期的な観点から積極的に取り組んでいく計画です。



最近の道路情況は、交通量の増大に伴ない、ちょっとしたトラブルや事故がもとで機能をマヒさせてしまいます。そこで円滑で安全な走行を維持する為に正確でスピーディな道路情報が要求されます。星和電機は道路情報のエキスパートとして、これらのニーズに応えるために時代を先取りした道路情報表示システムを提供しています。そしてこれが人と車のコミュニケーションをニューメディアで結ぶ、星和の安全ホットラインです。更にLED式表示装置の開発によりこの技術を応用したいろいろなLEDカラーディスプレーなど広範囲な利用方法の実用化をすすめています。また道路情報表示システム以外にもウォーターパールに代表される特殊照明器具やあらゆるシステムの動きの把握に不可欠な管理制御のルミグラフィック盤など、光学エレクトロニクス技術から生まれた偉大な機器たち。これらすべての開発には、はかり知れない情熱が寄せられています。つまりマンパワー。星和電機を支える力がそれです。そんな力をもっと集めたい、あえて言葉にするなら貴方がこれからの開発力。ニューパワーです。そのパワーを星和電機で形にしてみませんか。



●写真は道路情報表示装置、LEDカラーディスプレイ装置、道路照明、ルミグラフィックパネルの採用例です。

貴方がこれからの開発力。

●会社概要

設立　昭和24年1月
資本金　33億8000万円
株式上場　大証二部、京証
代表者　代表取締役　増山義三郎
従業員数　625名（男369名、女256名）
事業内容　道路情報表示装置、LEDマルチカラーディスプレイ装置グラフィック盤、産業用照明器具、配電制御機器などの製造・販売および設計・施工・請負
事業所
東京事業所/東京都千代田区神田和泉町1-12-17 久保田ビル
大阪事業所/大阪市西区靱本町1-12-6
名古屋事業所/名古屋市東区泉1丁目21-15
事業所/札幌、仙台、京都、広島、福岡、新潟、高松、静岡
売上高　217億円（93.3）
関連会社　星和電子㈱、㈲星和綜合

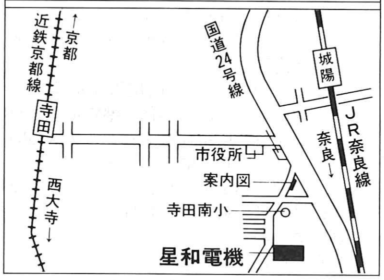
●募集要項

中途採用職種
①ソフトウェア開発エンジニア
制御・通信各種アプリケーション・ソフトウェアの開発
②電子回路設計エンジニア
デジタル・アナログ電子回路、論理回路の設計
③光半導体エンジニア
光半導体素子・アセンブルの設計、生産、応用（特にLED・液晶）
④ASIC開発エンジニア
マイコンソフト開発、ゲートアレイの開発
⑤ハイブリッドICエンジニア
ハイブリッドIC（デジタル・アナログ）の設計、生産
⑥製品検査エンジニア
各種制御・通信機器（コンピューター内蔵）の検査
⑦機構・構造設計エンジニア
制御・通信機器、監視盤等の製作設計、構造物強度計算
⑧屋外電気工事技術者（情報板の据付工事責任者）
本社、その他事業所

資格　学歴、性別問わず、経験者優遇
給与　当社規定により優遇
待遇　昇給年1回、賞与年2回、交通費全額支給、世帯主手当、資格手当、ほか
福利厚生　各種社会保険、住宅資金融資制度、財形貯蓄、社内貯金、社員持株制度、各種クラブ活動、独身寮
勤務地　本社または東京技術部
勤務時間　8:30～17:15
部門によりフレックスタイム導入

休日休暇　年間休日125日（日祝日・土曜〈完全週休2日制〉、年末年始、夏期休暇含む）有給休暇、特別休暇ほか
応募方法　履歴書（写真貼付）を郵送下さい。面接日を後日お知らせします。
●応募の秘密は厳守いたします。
●お気軽に電話にてお問合わせ下さい。
●入社日はご相談に応じます。
●担当/人事課

交通機関
本社/近鉄京都線寺田駅下車徒歩15分
JR奈良線城陽駅下車徒歩10分
東京事業所/JR秋葉原駅より徒歩5分
地下鉄秋葉原駅より徒歩3分

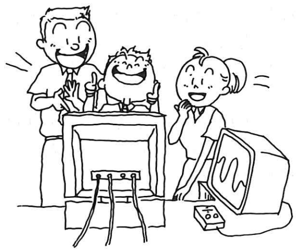


星和電機株式会社

〒610-01 京都府城陽市寺田新池36
☎(07745) 5-8181㈹

SAM

制御(コントロール)が得意(すき)。
情報処理(プログラミング)なら自信有(まかせておけ)。
機械語翻訳(アセンブラ)も操作(あやつり)、
取引先(ユーザー)の要求(ニーズ)を分析(ヒアリング)できる、
職人気質(スペシャリスト)の柔軟技術者(ソフトエンジニア)、募集(もとむ)。



SAMは,どんな仕事も技術的な面で断ることはない.ハード・ソフトのワクにはまらぬ職人気質の技術者たちが実に多種多様のシステムを作り出していく.自分の仕事にとことんこだわり続ける彼らには,大手企業から名指しで仕事の依頼が舞い込むこともある.頼りにされているという実感,新しい技術に出会う感動,それらが仕事に責任と誇りを与えてくれる.SAMは技術者の実力が100%発揮されることを第一に考え,気持ち良く働ける職場環境を目指している.

■会社プロフィール
設立／昭和51年1月　**資本金**／4000万円
代表者／篠宮　孝　**社員数**／35名
主要取引先／ソニー㈱、松下技研㈱、㈱日立製作所、㈱東芝、㈱ミツトヨ、山洋電気㈱、横河メディカルシステム㈱、他

■事業内容
●FA関連のマイコン応用電子機器システムの開発・設計・製作　●防災システムのマイコン応用検査治具システムの開発・設計・製作　●テレコミュニケーション関連　●ホトエレクトロニクス関連　●画像処理　●データ処理

■募集要項
職種・資格／別表参照
給与／経験・年齢・前給を考慮の上、当社規定により優遇します。
例）専門卒初任給　179,000円
大学卒初任給　199,000円
待遇／昇給年1回、賞与年2回(実績5ヵ月)、交通費全額支給、各種社会保険完備、退職金制度、家族会食会有、保養施設有り
勤務時間／9:00〜17:00(年間1700時間)
勤務地／本社(中目黒)
休日休暇／完全週休2日制(土曜、日曜)、祝日、年末年始・GW・夏期休暇
※年間休日122日、有給休暇
応募／電話連絡の上、履歴書(写真貼付)を持参下さい。郵送も歓迎。郵送の方には、追って面接日を連絡します。※応募秘密厳守
交通／中目黒駅下車徒歩6分、恵比寿駅下車徒歩9分
●資料請求される方は、巻末の資料請求ハガキ(飛竜企画宛)をご利用下さい。

職種	資格・経験	
ソフトウェア技術者	専門・大卒以上 20歳〜30歳位までの方	●パソコンおよびマイコン応用システムのソフト開発経験のある方歓迎(アセンブラ、C) ●また制御系、計測系のシステムを手がけてみたいと考えている方も、大歓迎です。
システムリーダー (技術管理者)	専門・大卒以上 25歳〜35歳位までの方 経験5年以上	制御系・計測系ソフト開発経験者 ●マイコン応用システムの開発経験者、ファームウェア開発経験者、歓迎します。 ●マネージメント経験者、優遇します。
ハードウェア技術者	専門・大卒以上 20歳〜30歳位までの方	●デジタル回路設計(CPU、DSP、VMEバス)および各種インターフェイスの経験者歓迎します。 ●アナログ回路設計できる方、優遇します。

SAM **エスエーエム株式会社**
〒153 東京都目黒区中目黒1-8-8 メグロF2ビル1F
☎03-5704-7811㈹　担当／篠宮

▶資料請求：TM09◀

派手な仕事じゃないけど、じっくり開発に取り組めるから、面白いと思う。



ひとつの製品の完成まで1～2年かける。
決して長いとは思いません。

とことん納得してから「商品」として世に出したい――。技術者なら誰でも、製品の開発には「こだわり」と「納得」を持ちたいと思うはずです。私たちニッシンエレクトロ㈱はこの「開発マインド」を崩すことなく、得意技術を中心とした製品開発にじっくりと取り組んでいます。製品の構想設計はもちろんのこと、仕様書の作成から試作品の製作、完成品まで、すべての工程を自分達で手掛けているため、製品の開発を通じて、自分たちの技術をじっくり伸ばしていける環境を生み出しています。

■会社概要■

設立 昭和46年6月
資本金 2,400万円
売上高 7億8000万円
従業員数 30名
事業内容 各種電子機器、制御機器の設計・開発・製造

■主要製品の開発実績■

- ●車輌交通量および車種判別計測システム
- ●鉄道用ATS装置
- ●光通信システム（LAN)
- ●オンライン情報伝送、およびテレックス等の回線制御装置
- ●電子交換機、特殊電話機用端末装置および試験装置
- ●業務用コンピュータおよびOA周辺機器
- ●フェンシング電気審判器
- ●IC化小型エンジン点火装置
- ●自動ダイヤル装置

■募集要項■

職種 電子回路設計技術者
ソフトウェア開発技術者
実装・機構設計技術者
資格 高卒以上、35歳位迄
給与 年齢・経験等を考慮の上、優遇いたします。
勤務地 本社（田無）
勤務時間 8：30～17：30
休日休暇 週休2日制、祝日、夏季・年末年始、年次有給休暇
待遇 昇給年1回、賞与年3回、各種社会保険完備、交通費全額支給、独身寮完備、保養所・レジャー施設各所、社員旅行
応募 電話連絡の上、履歴書（写真貼付）をご郵送、又はご持参下さい。追ってこちらからご連絡致します。
※応募の秘密厳守。
※入社日相談に応ず。
交通 西武新宿線田無駅下車徒歩15分

■主要取引先■

- ●松下電器産業株式会社
- ●松下通信工業株式会社
- ●株式会社ナカヨ通信機
- ●日本無線株式会社
- ●タナカ工業株式会社
- ●その他

■採用関係連絡先■

TEL.0424-65-9321㈹
担当／桑山・清水

●尚、8月7日～16日は、夏期休暇です。

勤務地田無／車通勤可

来春新卒者（高・専・大）同時募集

NISSIN **ニッシンエレクトロ株式会社** 本社・工場／〒188 東京都田無市芝久保町4-4-32
TEL.0424-65-9321㈹ FAX.0424-65-7255

▶資料請求：TM10◀

トランジスタ技術
インターフェース
合同広告企画 **Way** 技術者の求人情報

50音別広告索引

CQ Communication Card 希望の会社のハガキを使って資料請求をして下さい。巻末には、宛て先フリーの共通ハガキも付いています。手軽に企業にアクセスできる就職活動の第一歩として、ご活用下さい。

会社案内をご請求下さい

（掲載広告主が会社案内を用意しています。
ご希望の方は，広告主専用ハガキか宛先フリーハガキに，
下記の宛名及び裏面の必要事項をご記入の上，ご請求下さい。）

会社案内請求先宛名リスト

社名	〒	住所	所属	担当者
㈱コムシステム	550	大阪府大阪市西区江戸堀1-25-22 新江戸堀ビル		森田
山下システムズ㈱	146	東京都大田区池上5-7-12		久島
三幸電子㈱	468	愛知県名古屋市天白区植田3-802	総務	山口
アストロデザイン㈱	211	神奈川県川崎市中原区上小田中863	総務部採用担当	
㈱スリーデイ	145	東京都大田区南千束1-4-1 コーポートビル3F		新井
㈱インタフェース	732	広島県広島市南区南蟹屋1-1-25	本社総務部	森本
㈱金子製作所	177	東京都練馬区石神井台8-23-21	人事部	
㈱西湘エレクトロニクス	250	神奈川県小田原市南鴨宮3-45-4		瀬戸
イーデーエム㈱	173	東京都板橋区板橋3-5-2		塩原・神長
三基システムエンジニアリング㈱	150	東京都渋谷区渋谷3-6-2 第2矢木ビル		菊地
アルインコ㈱	540	大阪府大阪市中央区城見2-1-61 ツイン21・MIDタワー23F		
㈱ヒューモラボラトリー	167	東京都杉並区西荻北5-19-11		高橋・岩淵
㈱システムサコム	130	東京都墨田区両国4-38-16 両国桜井ビル4F	人事部	伊佐・当山
星和電機㈱	610-01	京都府城陽市寺田新池36	人事課	
エスエーエム㈱	153	東京都目黒区中目黒1-8-8 メグロF2ビル1F		篠宮
ニッシンエレクトロ㈱	188	東京都田無市芝久保町4-4-32		粂山
エル・エス・アイジャパン㈱	151	東京都渋谷区千駄ケ谷1-8-14	総務部	高田
㈱飛竜企画	153	東京都目黒区目黒2-11-11		

キリトリ線

会社案内請求カード

キリトリ線

郵便はがき

□□□-□□

41円切手をおはりください

御中

フリガナ 氏名	（　　歳）男・女
現住所または連絡先	〒 TEL
現在のお仕事	業種（　　　）職種（　　　） ※現在無職の方は，前職をご記入ください。
最終出身校又は在籍校	大学院・大学・高・専　学部　専攻
卒業年度	卒業・卒業見込

Way

トランジスタ技術
インターフェース 合同広告企画

全国の技術系読者に!

月2回(偶数月)

10日,25日発行





●広告料金(普通版)

4色1ページ	760,000
2色1ページ	600,000
白黒1ページ	470,000
白黒1/2ページ	300,000

お問合せは
☎03-5395-2132
CQ出版㈱ 広告部
C&E広告まで

キリトリ線

アンケートカード

1.当社について
イ.資料を送付してほしい
ロ.すぐ応募したい
ハ.人事担当者に会いたい(　月　日頃)

2.ご質問をご自由にお書きください

●ありがとうございました。

資料請求システム

▼積極的に技術者採用を行っている企業です。 興味を持たれた方は㈱飛竜企画行のハガキ〝資料請求サービス〟をご利用下さるか、直接企業へお問い合わせ下さい。

TM01 ㈱コムシステム
TM02 ㈱スリーディ
TM03 ㈱西湘エレクトロニクス
TM04 イーデーエム㈱
TM05 三幸電子㈱
TM06 三基システムエンジニアリング㈱
TM07 ㈱ヒューモラボラトリー
TM08 ㈱システムサコム
TM09 エスエーエム㈱
TM10 ニッシンエレクトロ㈱
TM11 山下システムズ㈱
TM12 三幸電子㈱

▼技術者採用企業情報です。上記企業と同様に資料請求サービスをご利用下さい。

S01	エキスパートマグネティックス㈱	〒275 千葉県習志野市津田沼4-11-11 小倉第一ビル TEL.0474-53-1655 担当／川井	FD・MD・光磁気ﾃﾞｨｽｸ等、記録媒体の検査装置を開発・製作・販売しています。世界中からの需要に対応中。技術者募集中。資料請求歓迎。
S02	㈱ソアーシステム	〒111 東京都台東区浅草橋1-9-12 秀山ビル TEL.03-5687-6171 担当／中村・佐久間	これからはメディアファンクションとそのアプリケーションの勝負。新しいハード／ソフトメディアの開発には夢と将来性がある。
S03	㈱マルチウェーブ	〒171 東京都豊島区池袋4-25-1紘亜ビル TEL.03-3985-9410 担当／斎藤	画像処理技術を応用したシステムを企画・開発。ハード・ソフト共に技術者募集中。
S04	㈱東亜プラン	〒167 東京都杉並区清水1-8-4東亜ビル TEL.03-5397-0511 担当／高野	電子応用ゲーム機器の企画・開発で急成長中。ハードウェアエンジニアを募集中。平均年齢27歳。
S05	ニッテツ北海道制御システム㈱	〒050 北海道室蘭市仲町12番地 TEL.0143-47-2793 担当／黒木	新日鉄㈱から分社独立。電子回路・機械設計技術者募集中。技術革新に燃える夢の探求者求む。
S06	㈱デュアルデザイナーズ	〒145 東京都大田区東雪谷2-21-14 TEL.03-3748-2955 担当／中村	マイコン応用技術と電子技術を融合し、ハード＆ソフトの企画・開発から設計・製造まで幅広く対応します。ハードとソフトの両方わかる技術者募集中。
S07	㈱システム開発	〒227 神奈川県横浜市緑区霧ヶ丘2-17-5 TEL.045-921-9311 担当／土屋	産業用オートメーション機器の開発設計（例：ＨＩＣ製造装置）。電子・機械の設計、調整ができる若手エンジニア募集中。一流大手企業が取引先です。
S08	新日本テクノシステム㈱ 労働大臣許可40-ｻ-0002	〒810 福岡市中央区天神4-9-10 第2正友ビル TEL.092-714-2630 担当／月俣	九州へのUﾀｰﾝ希望者へ。研究・開発・設計等の技術者を求めている企業を紹介します。費用一切不要。秘密厳守登録制度有。詳しい資料をご請求下さい。

キリトリ線

郵便はがき

料金受取人払

目黒局承認

793

差出有効期間
平成6年5月
31日まで

●切手不要

153-00

232

東京都目黒区目黒2-11-11

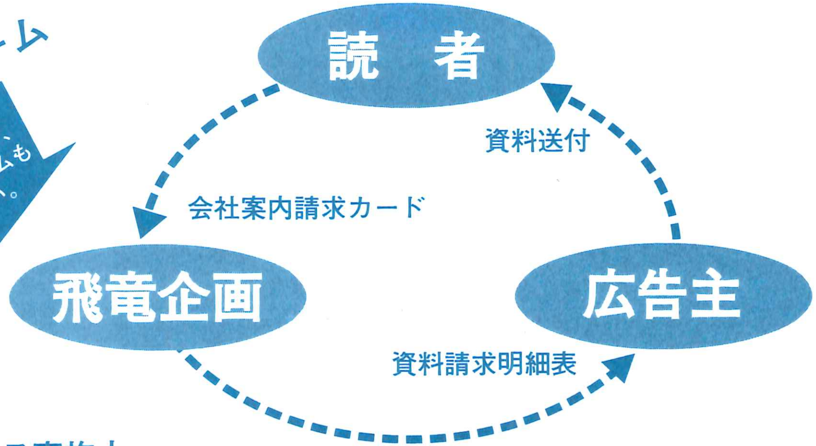
㈱飛竜企画内TM事業部
読者アンケート・資料請求係行

CQ 93-8

フリガナ 氏　名	（　　歳）男・女
現住所 または 連絡先	〒 TEL
現在の お仕事	※現在無職の方は、前職をご記入ください。 業種（　　　　）職種（　　　　）
最終出身校 又は在籍校	大学院 大　学 高・専　　　学部 専攻
卒業年度	卒業・卒業見込

資料請求システム

本誌掲載企業に興味をお持ちの方、資料請求システムもご利用下さい。



資料請求サービス実施中

本別冊付録の広告中、資料請求TMマークのついている企業に対し、資料請求サービスを行なっています。あなたが興味をお持ちになった企業をとじこみの(株)飛竜企画宛 Communication Cardでお知らせください。
こちらで広告主に連絡した後、広告主からあなたの元へ資料をお届けします。

掲載企業への資料請求は、このハガキをご利用下さい。

キリトリ線

資料請求サービスコーナー

1 本別冊付録の広告中で、あなたが興味を持った企業の資料請求番号をお書き下さい。
（但し、▶資料請求◀のある企業に限ります。）

NO.							

キリトリ線

アンケートにお答え下さい

1 資料請求の目的を○で囲んで下さい
- a．転職用資料として
- b．企業内容を詳しく知りたい
- c．取引先として検討したい
- d．その企業の製品やサービスについて知りたい

2 転職についてのご意見をお聞かせ下さい
- a．現在転職活動をしている
- b．希望する企業を捜している
- c．時期をみて、捜そうとしている
- d．考えているが、どの会社が良いかわからない
- e．具体的に考えていない
- f．その他（　　　　　　　　　　）

3 どちらのハガキをご利用になりましたか
- a．トランジスタ技術
- b．インターフェース

当社では積極的に人材の募集をおこなっています。
興味のある方はこのハガキをご利用下さい。

※ご協力ありがとうございました
秘密は厳守いたします

掲載企業への資料請求は、このハガキをご利用下さい。

※但し、▶資料請求◀のある企業に限ります。

ただいま、会議中。

MISでは、新人であれ、ベテランであれ、お互いに尊重しあう環境のなかでこそ、一人一人の創意工夫が活きてくると考えています。ワイワイ、ガヤガヤ…。個性豊かで、元気な発想がぶつかりあって、今日の会議もにぎやか。結論は予定より早く出そうです。



募集要項

職種▶ ●電子回路設計技術者
●ソフト開発技術者
●検査技術者
●品質・工程管理技術者
●市場開発、代理店管理

資格▶高卒以上20歳〜30歳位迄

勤務時間▶9:00〜17:30

勤務地▶大田区池上（大阪府中央区勤務希望の方はご相談に応じます。）

給与▶固定給制　経験・年齢を考慮の上優遇します。大卒初任給（'93年４月実績）／20.4万円（地域手当含む）

待遇▶諸手当／交通費全額支給、時間外手当、資格手当・昇給／年１回・賞与／年２回（'92年度実績平均6.0ヶ月）

休日▶完全週休２日制（土・日）、祝日、年末年始、慶弔、年次有給、リフレッシュ休暇

福利厚生▶各種社会保険制度、財形貯蓄制度、単身者向け家賃補助制度

会社概要

設立▶1973年11月

資本金▶1500万円

売上高▶5億3800万円（'92年度実績）

従業員数▶20名（男子12名、女子8名）

関連会社▶大阪エム・アイ・エス、山下電気、山下サーキテック、山下商事

事業内容▶マイクロコンピュータボードの設計、製造、販売

応　募

履歴書（写真貼付）を郵送して下さい。追って面接日を御連絡致します。

転職に関するお問い合せ歓迎▶将来転職をお考えの方もお気軽にお問い合せ下さい。（応募秘密厳守）

☎(03)5700-2121㈹ 担当 久島

〒146 東京都大田区池上5-7-12





交通：東急池上線池上駅より徒歩７分

misは皆さんの東京ライフを応援します。
仕事、暮らし、その他何でもご相談下さい。
フリーダイヤル
0120-440069

mis 山下システムズ株式会社